

TRABAJO DE FINAL DE CARRERA

16-1-2020

IOT EN EL SECTOR ENERGÉTICO:

SOFTWARE DE MONITORIZACIÓN Y
ANÁLISIS DE VARIABLES
AMBIENTALES Y ENERGÉTICAS.

GRADO EN INGENIERÍA DE LA ENERGÍA

Autor: Ricardo Boriba Makuere

Director: Joan Grau Barcelo

Co-director: Javier Farreras de la Morena

Resumen

El presente documento está dividido en dos bloques:

Bloque teórico: fundamentos teóricos del IoT para la fase experimental. En este bloque se explica todos los fundamentos teóricos necesarios para entender el concepto de Internet de las cosas o IoT como comúnmente se conoce y también se explica el conjunto de capas o tecnologías que se involucran en este concepto. Esta explicación teórica se hace con la finalidad de asentar las bases para poder entender mejor el segundo bloque que consiste en una fase más experimental.

Bloque Experimental: creación de la Prueba de Concepto (PdC). En esta parte del documento se crean dos pruebas de concepto, es decir, se implementa de manera resumida e incompleta lo que se ha explicado en el bloque teórico, con la intención de verificar que el concepto de IoT es susceptible de ser explotado de manera más útil.

La primera prueba de concepto creada se explica paso a paso y consiste en la creación de una página web en la que se visualiza mediante gráficos los datos capturados por un sensor de humedad y temperatura. Se trata de una página web sin servidor y utilizando tecnología de la computación en la nube y lenguajes de programación bastante conocidos como son Python, JavaScript, DynamoDB, etc.

La segunda prueba de concepto, consiste en incluir más datos de sensores (Luxómetro, medidor de energía, calidad del aire y presión) en la plataforma anterior. Al contrario que la web anterior, esta plataforma no se explica paso a paso, para limitar la extensión del documento, pero sí se provee de todos los recursos necesarios para su implementación.

A través de este link, puedes se puede acceder a ver el resultado final de las Pruebas de Concepto:

<http://iotcloudsolutions.com/>

Resum

El present document està dividit en dos blocs:

Bloc teòric: fonaments teòrics de l'IoT per a la fase experimental. En aquest bloc s'explica tots els fonaments teòrics necessaris per entendre el concepte d'Internet de les coses o IoT com comunament es coneix i també s'explica el conjunt de capes o tecnologies que s'involucren en aquest concepte. Aquesta explicació teòrica es fa amb la finalitat d'assentar les bases per poder entendre millor el segon bloc que consisteix en una fase més experimental.

Bloc Experimental: creació de la Prova de Concepte (PdC). En aquesta part de el document es creen dues Proves de Concepte, és a dir, s'implementa de manera resumida i incompleta tot el que s'ha explicat en el bloc teòric, amb la intenció de verificar que el concepte de IoT és susceptible de ser explotat de manera més útil.

La primera prova de concepte creada s'explica pas a pas i consisteix en la creació d'una pàgina web en la qual es visualitza mitjançant gràfics, les dades capturades per un sensor d'humitat i temperatura. Es tracta d'una pàgina web sense servidor i utilitzant tecnologia de la computació en el núvol i llenguatges de programació bastant coneguts com són Python, JavaScript, DynamoDB, etc.

La segona prova de concepte, consisteix a incloure més dades de sensors (Luxímetre, mesurador d'energia, qualitat de l'aire i pressió) a la plataforma anterior. Al contrari que el web anterior, aquesta plataforma no s'explica pas a pas, per limitar l'extensió de el document, però sí es proveeix de tots els recursos necessaris per a la seva implementació.

A través d'aquest link, es pot accedir a veure el resultat final de les Proves de Concepte:

<http://iotcloudsolutions.com/>

Abstract

This document is divided into two blocks:

Theoretical block: theoretical foundations of the IoT for the experimental phase. This block explains all the theoretical foundations necessary to understand the concept of the Internet of things or IoT as it is commonly known and also explains the set of layers or technologies that are involved in this concept. This theoretical explanation is made in order to lay the foundations to better understand the second block consisting of a more experimental phase.

Experimental Block: creation of the Proof of Concept (PdC). In this part of the document two proofs of concept are created, that is, what has been explained in the theoretical block is implemented in a summarized and incomplete way, with the intention of verifying that the concept of IoT is capable of being exploited in a more useful way.

The first proof of concept created is explained step by step and consists in the creation of a web page in which the data captured by a humidity and temperature sensor is visualized by graphics. It is a Serverless web page, using cloud computing technology and programming languages well known as Python, JavaScript, CSS, DynamoDB, etc.

The second Proof of Concept consists in including more data sensors (Luxometer, energy meter, air quality and pressure) in the previous platform. Unlike the previous website, this platform is not explained step by step, to limit the extension of the document, but it does provide all the necessary resources for its implementation.

Through this link, it is possible to view the final results of the Proof of Concep:

<http://iotcloudsolutions.com/>

ÍNDICE

RESUMEN	1
RESUM.....	2
ABSTRACT	3
1. OBJETO, ALCANCE Y MOTIVACIÓN	1
1.1. OBJETO.....	1
1.2. ALCANCE.....	2
1.3. MOTIVACIÓN	2
BLOQUE TEÓRICO: FUNDAMENTOS TEÓRICOS DEL IOT PARA LA FASE EXPERIMENTAL	3
1. INTRODUCCIÓN	4
1.1. ARQUITECTURA IoT.....	5
2. CAPA DE DETECCIÓN Y RECOLECCIÓN DE DATOS.....	7
2.1. SENSORES.....	7
2.2. ACTUADORES	10
2.3. CONTROLADORES.....	11
2.3.1. <i>El hardware de la Raspberry Pi (RPI)</i>	13
2.3.2. <i>El software de la RPi</i>	18
2.3.3. <i>Recopilación de la capa de detección y recolección de datos.</i>	23
3. CAPA DE TRANSMISIÓN DE DATOS.....	24
4. CAPA DE ALMACENAMIENTO Y CAPA DE ANALÍTICA DE LOS DATOS. CLOUD COMPUTING.	29
4.1. CAPA DE ALMACENAMIENTO DE DATOS	29
4.2. CAPA DE ANALÍTICA DE DATOS	31
4.3. AMAZON WEB SERVICES (AWS).	32
4.4. RESUMEN CAPA DE ALMACENAMIENTO Y CAPA DE ANALÍTICA DE LOS DATOS	34
5. CAPA DE PRESENTACIÓN Y VISUALIZACIÓN DE LOS DATOS.	35
5.1. LENGUAJES DE PROGRAMACIÓN UTILIZADOS.	36
6. SEGURIDAD Y PROTECCIÓN EN EL CONCEPTO DE IOT.	37
7. APLICACIONES IOT EN EL SECTOR ENERGÉTICO.	39
7.1. SCHNEIDER ELECTRIC Y MICROSOFT AZURE. IoT PARA LA ENERGÍA.	39
7.2. SENTILO TERRASSA Y SENTILO BCN.	40
7.3. DEXMA. DEXCELL: SOFTWARE PARA GESTIÓN ENERGÉTICA.	42
7.4. EMPRESAS Y STARTUPS DEL SECTOR IoT.....	43
8. CONCLUSIONES DE LA PARTE TEÓRICA	44
9. BIOGRAFÍA, REFERENCIAS Y RECURSOS UTILIZADOS DE LA PARTE TEÓRICA	45
BIBLIOGRAFÍA:	45
RECURSOS.....	45
BLOQUE EXPERIMENTAL: CREACIÓN DE LA PRUEBA DE CONCEPTO (PDC)	46
1. CONSTRUCCIÓN DE LA PLATAFORMA IOT.	47
2. PREPARACIÓN DE LA RPI COMO DISPOSITIVO IOT.....	48
2.1. INSTALACIÓN EL SISTEMA OPERATIVO A LA RPI.	48

2.2.	CONECTIVIDAD A INTERNET VÍA WI-FI.....	50
2.3.	CONECTIVIDAD A INTERNET VÍA CABLE ETHERNET	51
2.4.	COMUNICACIÓN CON LA RPI CON EL ORDENADOR PERSONAL VÍA SSH (SECURE SHEL) MEDIANTE PUTTY	51
2.5.	CONFIGURACIÓN BÁSICA DE LA RPI, PREPARACIÓN DEL ENTORNO VIRTUAL E INSTALACIONES DE TODO EL SOFTWARE NECESARIO.	56
3.	ESQUEMAS DE LOS CIRCUITOS Y CONEXIÓN DE SENSORES.	63
3.1.	CONEXIÓN E INSTALACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD DHT22.....	63
3.2.	CONEXIÓN E INSTALACIÓN DEL SENSOR DE BME680	66
3.3.	CONTRATIEMPOS SENSORES.	68
4.	PREPARACIÓN Y COMUNICACIÓN CON LA NUBE.....	68
4.1.	ARQUITECTURA BÁSICA DEL CONCEPTO IOT.....	68
4.2.	CONFIGURACIÓN DE AWS IOT.	70
4.3.	CONFIGURACIÓN DE DYNAMODB Y CREACIÓN DE REGLAS EN AWS IOT.....	82
4.4.	CREACIÓN DE LA PLATAFORMA WEB Y CONFIGURACIÓN DE S3.....	89
4.5.	CONFIGURACIÓN DEL DOMINIO CON ROUTE53.....	106
5.	COSTE DE CREACIÓN DE LA ARQUITECTURA BÁSICA.....	111
6.	ARQUITECTURA DEFINITIVA CON MÁS SENSORES.....	113
7.	5. CONCLUSIONES DE LA PRUEBA DE CONCEPTO (PDC)	116
8.	RECURSOS UTILIZADOS PARA CREAR LA PRUEBA DE CONCEPTO.....	117
	ANEXOS	118
	I. PLANIFICACIÓN DE TFG (DIAGRAMA GANT).....	118
	II. DESGLOSE DE COSTES AWS	118
	III. REPORTAJE FOTOGRÁFICO.....	118
	I. PLANIFICACIÓN DE TFG (DIAGRAMA GANT).....	119
	II. DESGLOSE DE COSTES AWS	122
	III. REPORTAJE FOTOGRÁFICO.....	129

1. Objeto, alcance y motivación

1.1. Objeto

El objeto del presente documento es dar a conocer un área de conocimiento relativamente reciente que va a transformar muchos sectores profesionales, con sectores profesionales me refiero a sector de la energía, salud, edificación y construcción, logística, etc. Esta nueva área de conocimiento es el Internet de las cosas o IoT como comúnmente se conoce.

La manera que se da a conocer el IoT es creando una web de visualización de los datos capturados por sensores y explicando los fundamentos teóricos necesarios para poder crear esta plataforma web, de esta manera explicar todo lo que abarca este concepto que actualmente se encuentra en el pico de expectativas infladas según el ciclo de sobre expectativa (Hype Cycle) de Gartner 2019.

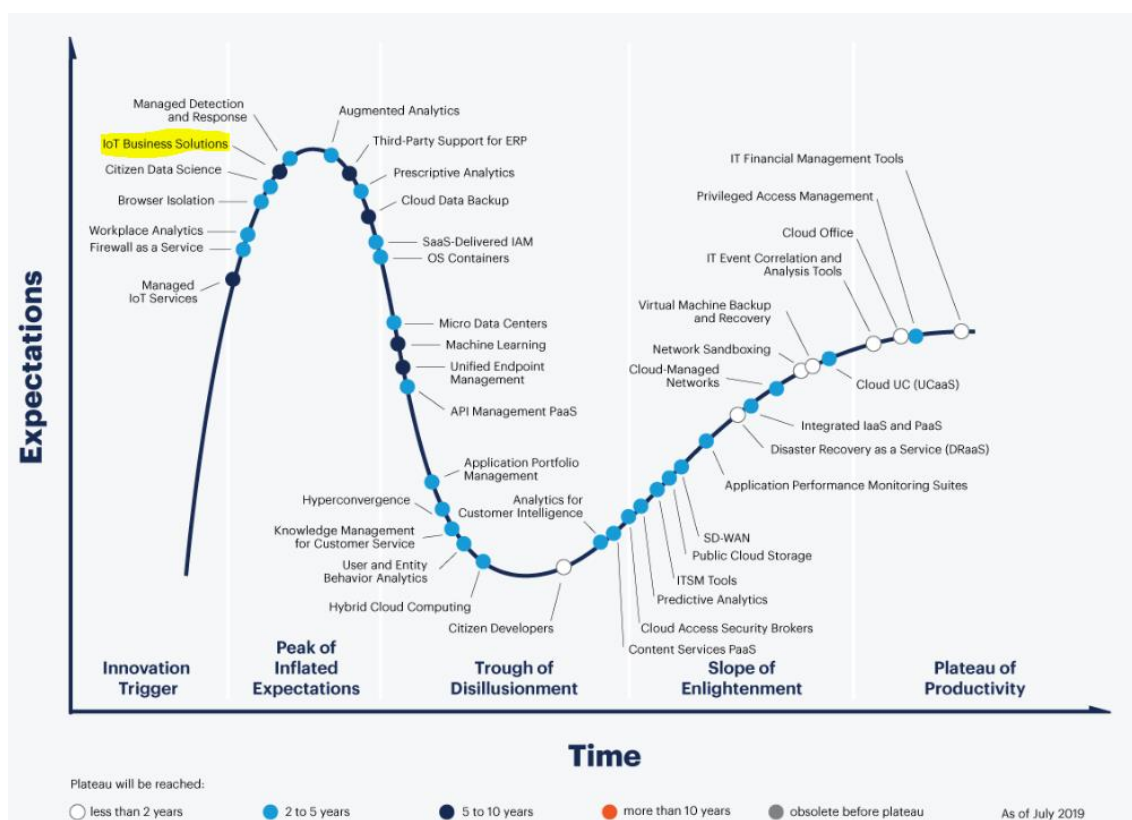


Figura 1. 1. IoT Hype Cycle. Fuente: [Gartner](#)

En el bloque teórico en concreto se intenta dar respuesta a las siguientes preguntas:

- ¿Qué es IoT? ¿Por qué es un concepto y no una tecnología?
- ¿Cuáles son las tecnologías que involucran el concepto de IoT?
- ¿Qué protocolo de comunicación utilizar?
- ¿Cuáles son los componentes Hardware y software para crear una prueba de concepto?
- ¿Qué es la computación en la nube? ¿Cómo aplicar los recursos que provee la computación en la nube para implementar una solución IoT?
- ¿Cómo encaja este nuevo paradigma con el sector energético o cualquier otro sector?
- ¿Existen empresas en la actualidad que utilicen el concepto de IoT como modelo de negocio?

1.2. Alcance

El alcance del presente documento es la creación y explicación de cómo construir una web básica sin servidor en el cual se muestre en tiempo real los datos capturados por diversos sensores.

1.3. Motivación

La motivación de realizar un proyecto como este es explorar una nueva área de conocimiento y de paso ir adquiriendo habilidades necesarias para responder a las necesidades reales del mercado, necesidades como programación, el cloud computing, electrónica, electricidad, etc. Ver cómo encaja este nuevo paradigma con los conocimientos adquiridos durante la carrera.

BLOQUE TEÓRICO: FUNDAMENTOS TEÓRICOS DEL IOT PARA LA FASE EXPERIMENTAL

1. Introducción

Internet De las Cosas (IdC) o más conocido en sus siglas en inglés IoT (Internet Of Things), es la interconexión digital de objetos de nuestro día a día a internet. Engloba el conjunto de tecnologías y servicios que facilitan la interconexión, transacción de la información, interacción y acción de elementos finales (objetos o cosas) que pueden ser dirigidos e identificados unívocamente con una dirección IP (Internet Protocol) [1].

Es importante esclarecer que se trata de un concepto y no de una tecnología. Pero para llegar a la descripción anterior serán necesarias gran variedad de tecnologías. Estas tecnologías pueden ser nuevas, para dar respuestas mejores a las necesidades de Internet de las cosas (IoT), o tecnologías ya existentes.

Denominamos Tecnologías IoT a aquellas tecnologías tanto nuevas como antiguas, que hagan realidad el concepto de Internet de las cosas.

Existen una gran variedad de conceptos con ligeros matices que apuntan todos a la misma idea de Internet de las Cosas. Estos conceptos son:

- Internet de todas las cosas o Internet de Todos (IoE, Internet Of Everytihngs).
- Industrial Internet Of Things (IIoT).
- Web de las Cosas (WOT, Web of Things).
- Internet de las personas y cosas.
- Machine-to-Machine (M2M, Máquina a máquina).
- Big Data
- Cyber-Physical-Systems (Sistemas ciberfísicos).

Los sistemas ciberfísicos (CPS) e IoT se utilizan indistintamente en algunos entornos, ya que los CPS son dispositivos conectados a internet, que nos permiten interactuar con el mundo físico. [1]

En los siguientes apartados se dan ejemplos más visuales, pero de momento para hacernos una idea, un ejemplo de una aplicación IoT en el entorno doméstico sería un sistema de calefacción capaz de adecuar por sí mismo la temperatura de confort de nuestro hogar según la temperatura del ambiente exterior sin intervención humana.

Este ejemplo nos lleva a los hogares inteligentes donde los dispositivos IoT son capaces de realizar una gran variedad de cosas como gestionar la eficiencia energética de nuestro hogar o incluso gestionar la seguridad. A pesar de que en el entorno doméstico existe espacio para el IoT, donde existe un espacio mucho más amplio donde crear y donde se está apostando por su desarrollo con mayor fuerza, es en la industria a gran escala. Industrias de gestión energética, salud, construcción o edificación, transporte o logística. IoT aplicado al sector industrial se conoce con el nombre de Industrial Internet Of Things (IIOT).

Recapitulando IoT permite conectar dispositivos a Internet, registrar, recopilar, recibir, enviar y comunicarse con otros dispositivos i/o aplicaciones con o sin intervención de personas mediante tecnologías IP, plataformas y soluciones de conectividad. Todo esto forma parte de un ecosistema más amplio de tecnologías como la inteligencia artificial, ciberseguridad, computación en la nube, Big Data y Blockchain entre otras. [1]

Los elementos principales de un ecosistema IoT se pueden representar según el siguiente diagrama:

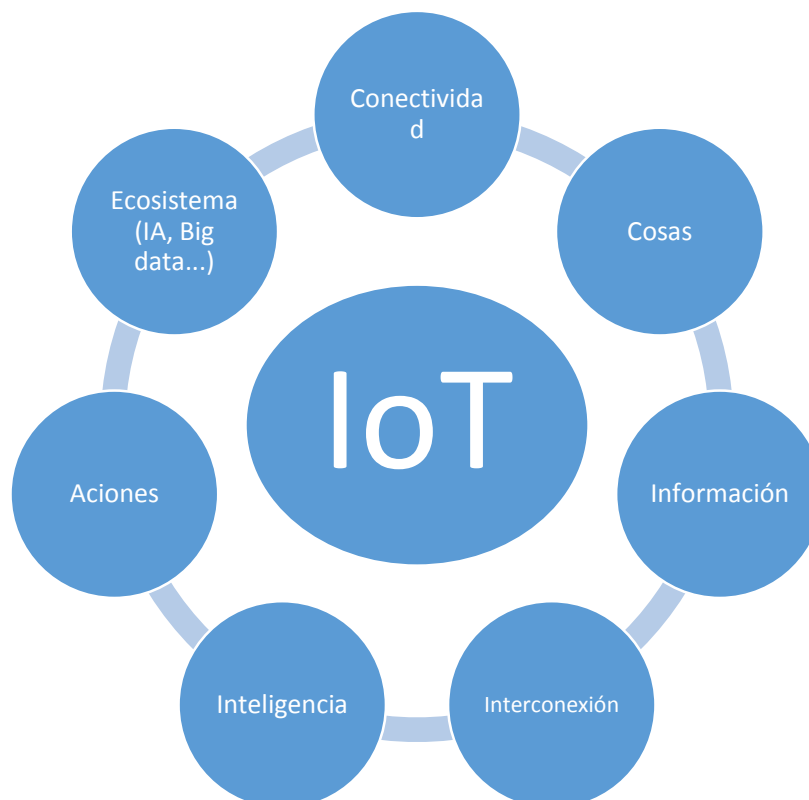


Figura 1. 2. Elementos principales de un ecosistema IoT. Fuente: [9]

1.1. Arquitectura IoT

Internet de las cosas consta de una serie de capas o componentes que configuran su arquitectura. Estas capas constan de una serie de tecnologías (Tecnologías IoT), servicios y protocolos. Las capas que hacen realidad el concepto de IoT son las siguientes [1]:

1. Capa de detección y recolección de datos.
2. Capa de transmisión de datos. Redes de comunicación, es decir, conectividad.
3. Capa de almacenamiento de datos en centros de datos (data centers). La nube o Cloud Computing.
4. Capa de analítica de datos.
5. Capa de presentación y visualización de los datos o de los resultados una vez tratados.



Figura 1. 3. Elementos principales de un ecosistema IoT. Fuente: [1]

A modo visual se muestra como sería una arquitectura genérica de un sistema IoT (Figura 1.4).

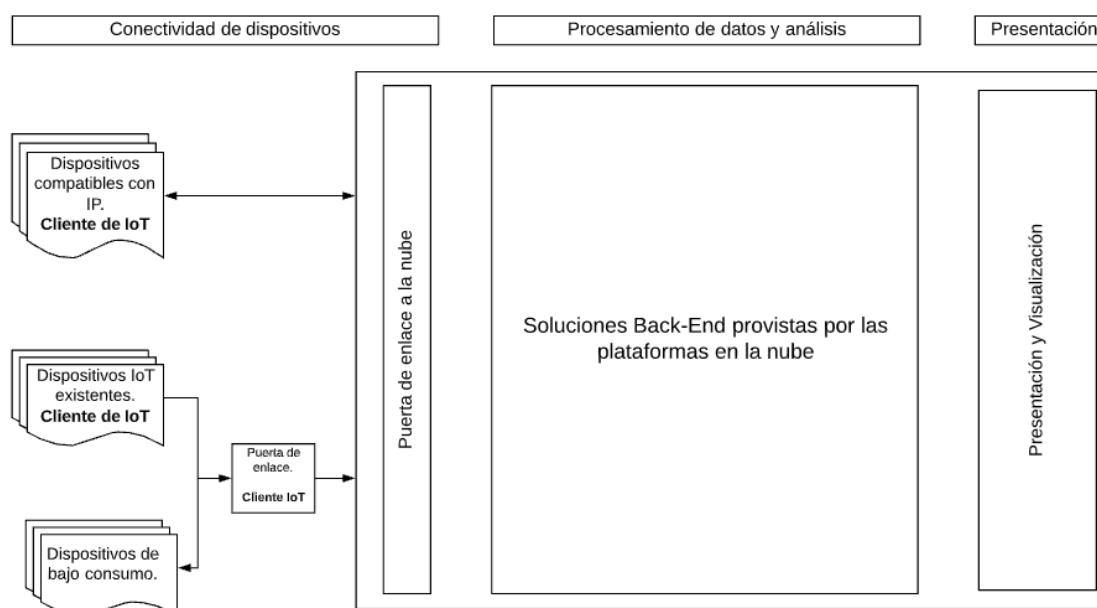


Figura 1. 4. Capas de una arquitectura genérica de IoT. Fuente: [1]

A lo largo del proyecto se utiliza esta arquitectura para ir definiendo cada una de las cinco capas. De esta es posible hacerse una idea mucho más detallada de todas y cada una de las tecnologías que hacen posible el concepto de Internet de las Cosas.

En los siguientes capítulos se darán a conocer los componentes hardware y software que se encargan de la detección y recolección de los datos (sensores, actuadores, PLC, etc); las diversas redes de comunicación que se encargan de la transmisión de datos (WAN, LAN, PAN, ...); la computación en la nube encargada de la capa de almacenamiento de datos, la capa de analítica de datos (AWS, Microsoft Azure, GCP, etc) que incluso a veces, también se encarga de la capa de presentación y visualización dependiendo de la arquitectura o solución a ofrecer.

El concepto de IoT abarca muchas tecnologías por lo que sólo se entra en detalle en aquellas tecnologías que se utilizarán en el segundo bloque del presente documento correspondiente a la parte de prueba del concepto (PoF) IoT, en la que se pone en práctica una arquitectura IoT básica con todo lo que se ha explicado.

A continuación, se explica detalladamente todas y cada una de estas capas que hacen posible el concepto de IoT.

2. Capa de detección y recolección de datos.

En esta capa, para la detección y captura de los datos se utilizan numerosos dispositivos como pueden ser sensores, controladores, actuadores y cualquier elemento susceptible de poder recoger información y recibirla, como pueden ser ordenadores, teléfonos inteligentes, chips RFID (identificación por radiofrecuencia), NFC (comunicación de campo cercano), QR (código de respuesta rápida), iBeacons, etc.

En la Figura 2.1, se muestra un diagrama que ayuda a ilustrar esta capa y la interacción entre el sensor, controlador y actuador.

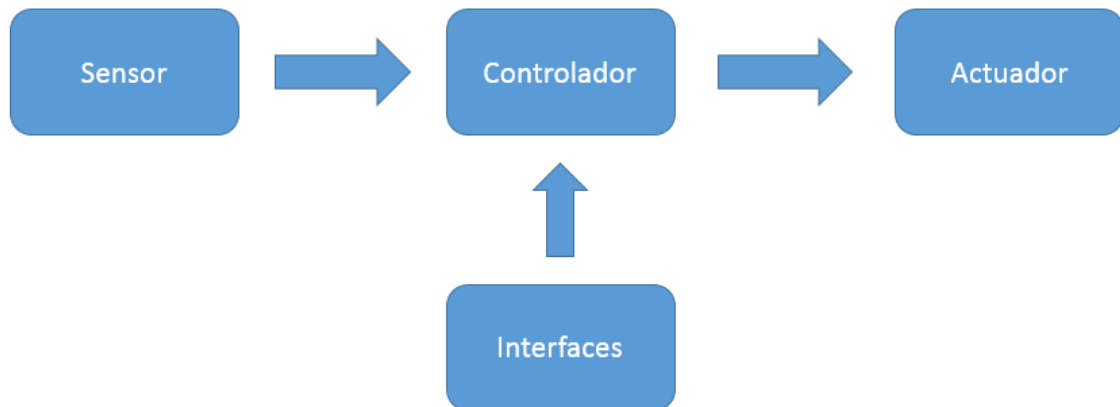


Figura 2.1. Diagrama de la capa de detección y recolección de datos. Fuente: [1]

2.1. Sensores

Los sensores son dispositivos físicos que conectan con el entorno, recogen información de diferentes tipos y la transforman en una señal eléctrica que puede ser digital o analógica.

Las señales digitales, son de todo o nada, es decir, tienen únicamente dos posibles valores, que detectan simplemente si llegamos a un determinado valor de la variable del entorno. Estos sensores se conocen como detectores. [2]

Las señales analógicas, disponen de un rango continuo de valores, que equivalen a un rango de medida de la variable física, por ejemplo, de 0 a 100 °C.

Los sensores, sean analógicos o digitales, en el entorno de tecnologías IoT que nos ocupa, pueden clasificarse como:

- **Detectores:** detectan un determinado valor. Pueden ser:
 - **Temperatura:** nos permite controlar que se alcanza o baja de una determinada temperatura, como los termostatos.
 - **Recursos:** permite conocer si se llega a un determinado valor, como es el caso de los detectores de nivel de líquidos y sólidos.
 - **Seguridad:** generan un aviso cuando se detecta un determinado fenómeno físico o se alcanza un determinado valor. Ejemplos de este tipo de detectores son los detectores de humo o los detectores de gas.
 - **Iluminación:** permiten conocer si hay un determinado nivel de iluminación en un ambiente.

- **Presencia o movimiento:** los detectores de movimiento o los detectores de intrusión son un buen ejemplo. Estos detectores nos permiten detectar el movimiento o la presencia de una persona.
- **Sensores simples:** recogen una sola medida o un conjunto determinado de medidas. Pueden ser:
 - **Recursos:** nos permiten conocer el consumo (contadores de energía) o reservas (sensores de nivel).
 - **Seguridad:** ejemplos de este tipo de sensores son los detectores de humo, de contaminación, de intrusión, etc.
 - **Iluminación:** nos permiten conocer el nivel de iluminación de un ambiente, como es el caso del luxómetro.
 - **Presencia o movimiento:** nos permiten detectar un movimiento o la presencia de una persona. Al ser sensores y no detectores, estos están dotados de doble tecnología (movimiento y presencia) y se comunican mediante protocolos de seguridad.
 - **Condiciones climatológicas:** equipo o conjunto de sensores que miden la temperatura, la velocidad del viento, la humedad, etc.
 - **Ruido:** sensor que detecta la intensidad del sonido de un ambiente, sin reconocer voz ni frecuencias específicas.
- **Sensores complejos:** recogen información de un gran número de aspectos. Pueden ser:
 - **Infraestructura de transportes:** sensores de presencia (cámaras, infrarrojos, etc), de contaminación, radares de velocidad y sistemas de identificación de vehículos, etc.
 - **Movimiento:** en este caso ofrecen información sobre el movimiento de un objeto en diversos planos, rotaciones, etc.
 - **Posición:** los GPS (Sistemas de Posicionamiento Global), son un buen ejemplo. Estos sensores ofrecen la dirección de la componente horizontal del campo magnético natural.
 - **Estaciones meteorológicas y sistemas de seguimiento solar:** recogen información del nivel de radiación solar, posicionamiento del sol, valores de temperatura ambiente, humedad relativa, velocidad del viento, todo con un mismo aparato sensor.
 - **Seguridad:** este tipo de sensores incluyen las cámaras con reconocimiento de imagen, dirección, etc.
- **Sensores de identificación:** son sensores que permiten la identificación de un objeto incorporando información de propio objeto. Pueden ser:
 - **Etiquetas RFID (identificación por radiofrecuencia):** es un dispositivo que se puede adherir a cualquier cosa, a un libro, por ejemplo. Estos dispositivos contienen antenas que permiten recibir y responder a peticiones por radiofrecuencia desde un emisor-receptor RFID. Se utilizan con más frecuencia en la gestión de inventarios.
 - **Etiquetas NFC (comunicación de campo cercano):** es una tecnología de comunicación inalámbrica de corto alcance y alta frecuencia, permitiendo de esta manera el intercambio de datos entre dispositivos. En la actualidad se puede ver con más frecuencia que se realizan pagos con los teléfonos

inteligentes, esto es debido a que los nuevos modelos incorporan esta tecnología.

- **Códigos BiDi y QR (código de respuesta rápida):** son códigos que contienen información codificada, y que permiten consultar información ampliada sobre múltiples objetos y elementos. Son muy parecido a los códigos de barras, donde estos incluyen información que puede consultarse a través de nuestros teléfonos inteligentes.

- **Dispositivos de captura de datos complejos.** Son elementos de captura de datos mediante cámaras o lectores, por ejemplo, las pistolas de contaje.

Después de esta larga lista se puede llegar a interpretar que hay una gran variedad de sensores que se encuentran por todas partes y en nuestro día a día.

A modo ilustrativo se muestran las siguientes imágenes de sensores tanto del sector industrial como el sector Do It Yourself (DIY) de los makers:

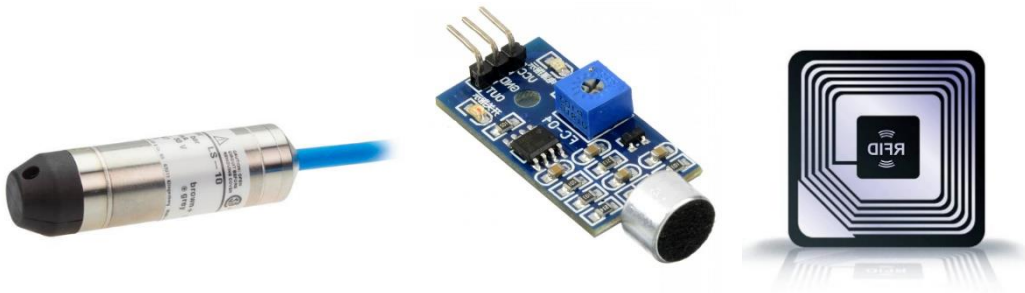


Figura 2.2. De izquierda a derecha: sensor de nivel industrial, sensor de ruido, RFID. Fuente: google imágenes







Figura 2.2. De izquierda a derecha: sensor NFC, sensor de temperatura industrial. Fuente: google imágenes



Figura 2.3. De izquierda a derecha: código QR, sensor de presencia, sensor de movimiento. Fuente: google imágenes

A continuación, se dan a conocer los detalles de los sensores utilizados:

Tabla 1. Sensores utilizados en la prueba de concepto (PdC). Fuente: elaboración propia

Imagen del sensor	Descripción
	Módulo DHT-22 Sensor Digital de Temperatura y Humedad AM2302 para Arduino o Raspberry Pi
	Breakout BME680 - Calidad del aire, temperatura, presión, sensor de humedad. Compatible con Raspberry Pi.
	Medidor de energía monofásico con protocolo de comunicación Modbus, viene incluido una pinza amperimétrica. SDM120CT-MV + ESCT-TU16 50A/0,1 V 1 de la Fase 2 de 230V RS485 Modbus y medida de kwh kvarh U-P Q PF Hz dmd medidor de energía de carril DIN.
	TSL2591 – Sensor de intensidad de luz digital – luxómetro

2.2. Actuadores

Los actuadores son elementos que transforman una señal digital o analógica, o una orden recibida a través de un bus de datos, en una acción. Normalmente son de tipo sí/no, como puede ser, un termostato que detecta el estado de la temperatura y decide encender o no la máquina de climatización. [2]

Los actuadores no están limitados al mundo de los elementos mecánicos y eléctricos (motores, válvulas, compresores, etc.) sino que también pueden ser dispositivos completamente digitales y realizar acciones en el mundo digital [2]. Un sistema que envíe un mensaje al teléfono móvil cada vez que tengas que regar una planta, sería un ejemplo.

Los actuadores pueden ser de tipo:

- **Todo o nada:** son actuadores de señal digital, como los relés o contactores, que permiten, por ejemplo, encender o apagar un grupo de lámparas, o cerrar una llave de paso de agua.
- **Regulación:** son actuadores de señal analógica, que generan una salida analógica proporcional. Esto permite, por ejemplo, regular el nivel de luz de un lugar o el caudal de una batería.
- **De bus:** son actuadores que, al recibir una orden por un bus de comunicaciones y un protocolo determinado, pueden generar una actuación tanto de tipo todo o nada como de regulación.

A modo ilustrativo se muestran las siguientes imágenes de actuadores del sector industrial, del sector de la domótica o el sector de los Makers:

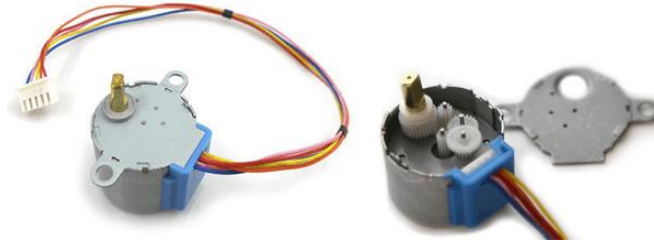


Figura 2.4. Motor DC Arduino. Fuente: google imágenes



Figura 2.5. De izquierda a derecha: relé, contactor, relé de mando. Fuente: google imágenes

2.3. Controladores

Son los dispositivos que se encargan de recoger las señales de los sensores o detectar y enviar las ordenes a los actuadores. También se encargan de enviar reportes de la información recogida por los sensores a la plataforma de servicios (capa de presentación y visualización de datos) y recibir órdenes de la propia plataforma para ser ejecutada por los actuadores.

Existen equipos remotos, que dependen de las órdenes de la plataforma de servicios para realizar cualquier acción, y equipos autónomos, que no dependen de la plataforma para realizar cualquier actuación. [2]

Hay situaciones en las cuales no es necesario un equipo controlador y resulta ser que los actuadores y/o sensores están dotados de suficiente inteligencia como para ser controlados directamente desde la plataforma de servicios.

Los algoritmos no solo se presentan como dispositivos hardware, también pueden ser de tipo software, como los drivers de la impresora que permiten su funcionamiento o las aplicaciones de nuestros dispositivos móviles que nos permiten realizar una gran variedad de acciones.

A continuación, se detalla los principales grupos de equipos que pueden realizar las funciones de controlador.

- **Registrador de datos:** este es el grupo de los dataloggers. Dispositivos que almacenan la información recibida por los sensores y detectores, con la fecha y la hora en que se han producido. Estos dispositivos solo realizan la función de aportar información.

- **Autómatas programables:** son equipos electrónicos que en función de cómo ha sido programado, actuará sobre la instalación dependiendo de las señales recibidas por los sensores. Estos equipos se conocen más como PLC (Programmable Logic Controller) y si disponen de reloj en tiempo real pueden hacer la función de dataloggers.
- **Unidad de control local:** son equipos eléctricos que disponen de una programación específica que se puede configurar para que actúen sobre una instalación de una determinada manera dependiendo de las señales recibidas por los sensores. Por lo general disponen de reloj en tiempo real, para realizar funciones de registro de datos.

A modo ilustrativo se muestran las siguientes imágenes de controladores:



Figura 2.6. De izquierda a derecha: Data Loggers y un PLC LOGO de siemens. Fuente: google imágenes

La Raspberry pi no es un controlador, actuador o sensor, es más bien un ordenador de una sola placa y aquí en el proyecto se utiliza como si fuese un dispositivo IoT Gateway o una pasarela de comunicación IoT.

Ordenadores de una sola placa: (en inglés: Single Board Computer o SBC) son computadores compuestas de un solo circuito. Ejemplo, de estos ordenadores son Arduino y la raspberry Pi.

IoT Gateways. Nuevas soluciones tecnológicas que están surgiendo en el sector industrial para permitir las comunicaciones IoT, normalmente comunicaciones dispositivo a dispositivo o comunicaciones dispositivo a la nube. Se trata de un hardware que aloja un software que realiza las tareas esenciales. Una de las funciones más básicas es la de facilitar la conexión entre diferentes fuentes de datos y su destinación.

También existen soluciones IoT en el que se utilizan combinaciones de dos o más controladores como puede ser la raspberry Pi junto con un autómata programable.

En el presente proyecto se utiliza la Raspberry Pi (RPi), para que realice las funciones de controlador, por lo que en los siguientes sub-apartado se entra en detalle en todos los aspectos necesarios para poder entender un poco más el funcionamiento y manejo de la RPi.

2.3.1. El hardware de la Raspberry Pi (RPi)

La Raspberry Pi forma parte del conjunto de ordenadores de una sola placa como ya se ha mencionado anteriormente. A diferencia de Arduino la RPi es un ordenador completo capaz de soportar un sistema operativo.

Existen muchas versiones de la RPi (RPi 3, RPi 2, RPi B+, RPi Zero, etc) y en función de nuestras necesidades para el prototipado nos convendrá una versión u otra. En el presente documento se utiliza RPi 3 Modelo B+ con las siguientes características:

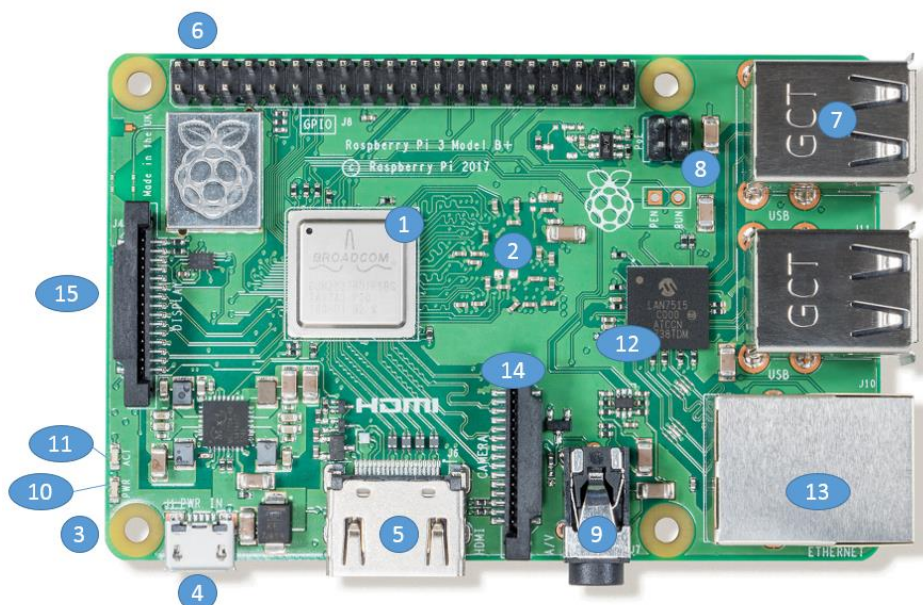


Figura 2.7. Las entradas/salidas y subsistemas de la RPi 3 B+. Fuente: Elaboración propia

Tabla 2. Descripción de las entradas/salidas y subsistemas de la RPi 3 B+. Fuente: [3]

Función	Dispositivos	Detalles
Procesador (1)	BCM283x(CPU)	Las placas RPi utilizan procesadores BCM2835/BCM2836/BCM2837 de Broadcom. Las distintas placas usan modelos ligeramente diferentes, con frecuencias que van de los 700 MHz a 1,2 GHz. Asimismo, están basados en núcleos de procesador ARMv6, ARMv7, ARMv11, ARMv8 A53.
	Motor gráfico (GPU)	Subsistema gráfico 3D Broadcom VideoCore IV 3D con controlador OpenGL ES1.1 Y 2.0.
Memoria (2)	256 MB a 1 GB DDR	La cantidad de memoria del sistema afecta al rendimiento y el uso del RPi como plataforma de computación de propósito general. La memoria se comparte entre la CPU y la GPU.
Almacenamiento (3)	Tarjeta Micro-SD	Todas las placas RPi se inician desde una tarjeta SD o micro-SD, con la excepción del modelo Compute Module. Ésta tiene soldado un módulo eMMC, que en efecto es una tarjeta SD en un chip. El RPi 3 utiliza una ranura de ajuste por fricción, en lugar de las habituales de encaje por "clic".
Alimentación (4)	Conector micro-USB	Se necesita una tensión de 5 V que, idealmente, aporte al menos 1,1 A (2,5 A para el RPi 3). Esta entrada cuenta con protección para sobretensiones.

Vídeo (5)	Conector HDMI o mini-HDMI	Se utiliza para conectar las placas RPi a un monitor o televisión. Los modelos RPi soportan hasta 14 resoluciones de salida, incluida la full-HD (1920x1080) y 1920x1200.
GPIOs (6)	Cabecera GPIO de 40 (o 26) patillas	40 patillas multiplexadas para proporcionar acceso a las características listadas en las siguientes filas de la tabla. No toda la funcionalidad está disponible al mismo tiempo.
	26 patillas GPIO	Patillas de E/S de propósito general que sirven para la lectura/escritura de datos binarios. El número máximo de patillas GPIOs, es de 16 en los modelos de 40 patillas. Todas las GPIOs toleran voltajes de 3,3 V. El uso de buses y otras interfaces reduce los GPIOs disponibles.
	2 buses I ² C	El I ² C es un bus digital que nos permite conectar varios módulos a cada uno de los buses de dos hilos al mismo tiempo. Uno de estos dos buses está reservado para su uso con placas HAT.
	Bus SPI	El bus SPI (interfaz de periféricos serie) proporciona un enlace serie síncrono de datos digitales para distancias cortas. Utiliza una configuración maestro/esclavo y necesita 4 hilos para la comunicación. El bus SPI del RPi tiene soporte Linux para dos líneas esclavas.
	UART	Utilizado para las comunicaciones serie entre dos dispositivos. El RPi, menos el RPi 3, suele tener un UART asignado de forma predeterminada para ofrecer conexión de consola serie.
	PWM	Las salidas de modulación por ancho de pulsos (PWM) nos permiten enviar un tipo de salida analógica que puede usarse para controlar diferentes dispositivos, por ejemplo, motores. Existe al menos una salida hardware PWM en todas las placas RPi, y dos en los modelos más recientes.
	GPCLK	Los relojes de propósito general (CPCLK) nos permiten establecer señales de temporización precisas.
Hub USB (7)	Puerto USB	Hay un hub USB interno en todos los modelos RPi con un número de puertos diferentes. Por ejemplo, el RPi 2/3 tiene 5 puertos USB internos, uno está conectado al puerto Ethernet y los otros cuatro están disponibles para conexiones externas.
Reinicio (8)	RUN desocupado	Se puede utilizar como botón de reinicio para el RPi.
Audio y vídeo (9)	Jack cuatripolar 3,5mm	Este conector ofrece vídeo compuesto, así como vídeo y audio estéreo en las placas más modernas.
LED Alimentación (10)	PWR LED	Indica que la placa está encendida. (No en la RPi Zero)
LED Actividad (11)	ACT LED	Indica que la placa está realizando algún tipo de actividad.
USB a Ethernet (12)	SMS LAN9514	Proporciona un hub USB 2.0 y un controlador Ethernet 10/100 BaseT. Las placas RPi se conectan a Internet por USB en lugar del controlador Ethernet incorporado en el SoC.
Red (13)	Ethernet RJ-45	Conector Ethernet RJ-45 que proporciona acceso Ethernet 10/100 Mbps. El RPi 3 cuenta con WiFi y Bluetooth en placa mediante el chip BCM43438.

Cámara (14)	CSI	El RPi tiene una interfaz CSI que cumple el estándar MIPI, que implementa en un zócalo especial 15 patillas que se puede emplear para conectar una cámara especial para el RPi.
Pantalla (15)	DSI	El RPi tiene una interfaz DSI (Interfaz serie para pantalla), como la que se suele utilizar en los teléfonos móviles con las pantallas táctiles. Hay pocas pantallas disponibles que soporten esta interfaz para el RPi.

Se he elige este modelo porque en su momento era el más completo y dispone de WiFi y Bluetooth integrado. Tecnologías de comunicación adecuadas para Internet de las Cosas. En la actualidad existe la RPi 4, con características aún mejores.

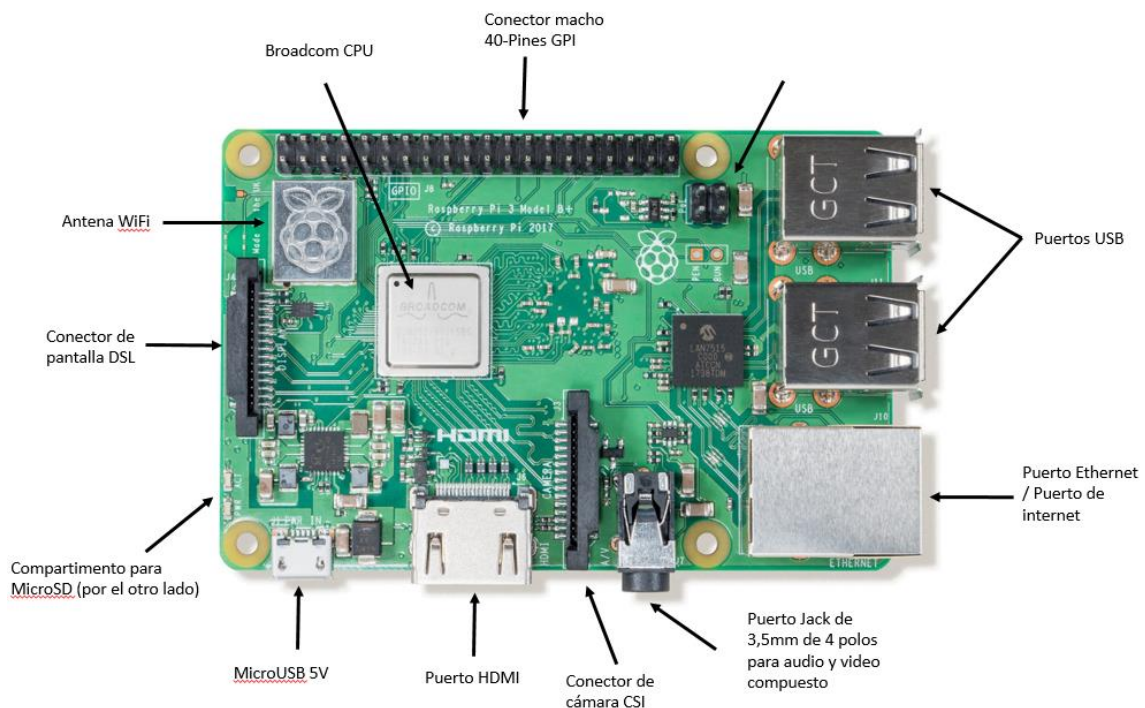


Figura 2.7. Imagen de la RPi Modelo B+3. Fuente: Elaboración propia

Tabla 3. Características técnicas de la RPi 3 B+. Fuente: manual del producto

RPi 3 Modelo B+	
Arquitectura	ARM v8-A 64-bit
SoC	Broadcom BCM2837B0
CPU	1.4 GHz 64-bit quad-core ARM Cortex-A53
RAM	1GB LPDDR2 SDRAM
Wireless LAN	2.4GHz and 5GHz
Bluetooth	Bluetooth 4.2 and BLE
Supports PoE	✓ (requieres un PoE HAT separado)
GPIOs	40
Puertos	HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

Accesorios de la (RPi).

Para comenzar a utilizar la Raspberry Pi se necesitan como mínimo los siguientes componentes:

- Fuente de alimentación. Adaptador de corriente de 5V 2.5A/3A con interruptor.
- Una tarjeta micro-SD para guardar el sistema operativo, necesario para iniciar la placa.
- Un cable Ethernet CAT-5 para conectar la RPi a la red mediante conectar Ethernet RJ-45 10/100.

Existen muchos accesorios como pueden ser un cable HDMI para conectar a un televisor, teclado, ratón, webcam y un sinfín de accesorios que nos permiten crear un ordenador mucho más completo. También existen accesorios para proteger la RPi, como el disipador de calor o la carcasa protectora diseñada especialmente para la Raspberry Pi.

Para el proyecto que nos ocupa, utilizaremos lo justo y necesario. Prescindiremos de pantalla, teclado y ratón. Los equipos que no utilizan estos componentes se conocen como Headless computers.

En la siguiente tabla se hace una comparación entre utilizar la RPi en modo Headless vs “GUI mode”, que significa utilizar la RPi con todos los componentes típicos de un ordenador. Y de esta manera hacerse una idea de las ventajas que presenta esta manera de utilizar la RPi para realizar el proyecto.

Tabla 4. Headless Mode VS GUI Mode. Fuente: curso RPi

Headless Mode	GUI Mode
Sin monitor, ratón ni teclado	Requiere monitor, ratón y teclado
Sistema operativo ligero	El sistema operativo es más pesado, para permitir los componentes gráficos.
Funciona utilizando otro ordenador vía conexión remota	Puede utilizarse directamente, no necesita un ordenador separado.
Útil para servidor o aplicaciones remotas	Útil para aplicaciones de escritorio
Backup y actualizaciones mucho más rápidas.	Backups y actualizaciones mucho más largas.
Fiabilidad mejorada	Fiabilidad aceptable
Rendimiento mejorado, RAM se dedica a las aplicaciones	La RAM se comparte con el video y las aplicaciones.
Debes estar familiarizado con la línea de comandos.	Los usuarios pueden apuntar y hacer clic en la interfaz.

Concretamente se utilizan, los siguientes elementos:

- Tarjeta micro SD SanDisk Clase 10 de 32 GB.
- El adaptador de corriente con un conveniente interruptor de on /off que proporciona energía eléctrica de 5V 3A.
- Carcasa negra de buena calidad para Raspberry Pi 3 B+ con fácil acceso a todos los puertos, cubierta removible para acceder a los GPIOs y abertura de ventilación.
- Disipador de calor de cobre selecto.
- Cable Ethernet Cat 5E o Cat. 6

Al disponer de WiFi integrado en la propia RPi, no será necesario un cable Ethernet, aunque viene bien disponer de uno para las configuraciones iniciales.

Buenas Prácticas de uso.

El microprocesador de la RPi no se puede sustituir, si se daña el SoC (System on Chip) del microprocesador, será necesario comprar otra RPi. Debido a esto y entre otras cosas más hacen que la Raspberry pi sea un dispositivo delicado el cual requiere que seamos diligentes en su manejo.

Como prácticas de buen uso, se listan aquí cosas que no se deben hacer [3]:

- No apagar la Raspberry Pi desenchufando directamente el cable de alimentación USB de la placa. Esto es como si se fuera la luz mientras te encuentras utilizando el ordenador de sobre mesa. Es mejor utilizar el procedimiento de pagado por software y después desconectar de la fuente de alimentación.

>> \$ sudo shutdown -h now

- No colocar la RPi conectado a la fuente de alimentación sobre superficies metálicas. Debido al calentamiento de la RPi, esta puede llegar a cortocircuitar los pines de la cabecera que sobresalen por la parte inferior de la placa llegando a inutilizarla. Se recomienda utilizar una carcasa o ponerlo en una superficie de madera.
- No conectar circuitos source/sink (impulso/drenaje de corriente continua) salvo corrientes muy pequeñas, desde/hacia la cabecera GPIO.
- Los pines GPIO toleran 3,3 V. No conecte circuitos alimentados a 5 V o inutilizará la placa.
- No conecte circuitos que apliquen potencia a la cabecera GPIO en tanto el RPi no esté encendido. Asegurarse de conectar todos los circuitos autoalimentados por la línea de 3,3 o mediante octoacopladores.

Para evitar errores a la hora de realizar conexiones de los diferentes sensores, es necesario conocer adecuadamente la cabecera GPIO que tiene 40 patillas y saber diferencia entre el sistema de numeración física de los pines y el sistema de numeración Broadcom o GPIO/BCM System de los pines. A continuación, se muestra una imagen que facilita la comprensión a la hora de realizar las conexiones con los sensores.

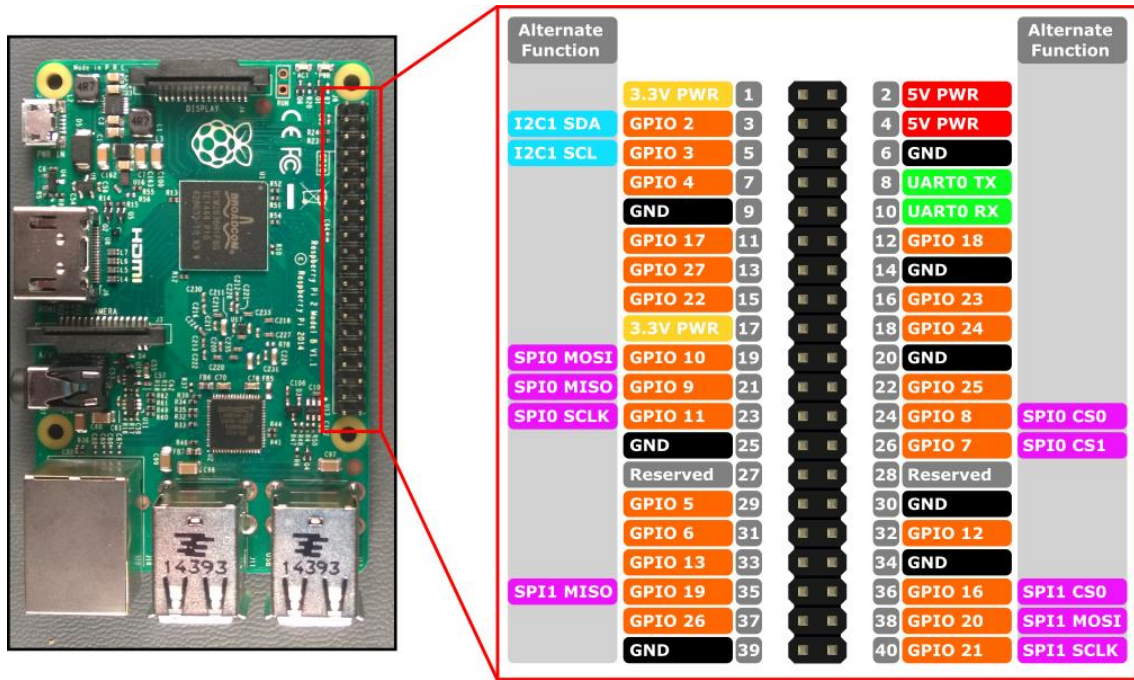


Figura 2.8. Nombres identificativos de los Pines de la RPi 3 B+. Fuente: google imágenes

En los próximos apartados, se explica todos los aspectos necesarios en relación con el software necesario para entender cómo realizar la comunicación con la Raspberry Pi para programarla o realizar cualquier actividad que deseemos.

2.3.2. El software de la RPi.

En este apartado se da a conocer los conocimientos básicos sobre el sistema operativo Linux, la conexión a internet a la RPi, sobre cómo comunicarse con la RPi en modo Headless y por último conocer los comandos básicos del sistema operativo Linux para realizar todo lo necesario para el proyecto.

Linux en la RPi

Linux distributions es una versión de dominio público de Linux que contiene un conjunto determinado de herramientas y programas (Red Hat Enterprise, Ubuntu, Debian, etc). Todas estas distribuciones comparten el mismo núcleo (*kernel*) estándar de Linux [3].

Las principales distribuciones de Linux (Sistemas Operativos) para ordenadores personales que son utilizadas por la comunidad son Raspbian, Ubuntu, OpenELEC y Arch Linux.

Raspbian es una versión de Debian (otra distribución de Linux) creada especialmente para la Raspberry Pi, por lo que utilizaremos este sistema operativo para realizar la prueba de concepto.

Existen otros sistemas operativos creados para poder funcionar con la RPi, sin problemas, creadas por otras empresas, por ejemplo, el Windows 10 IoT de Microsoft.

A lo largo de los años se han ido creando distintas versiones de Raspbian como son Raspbian Jessie y Raspbian Wheezy. En el momento de escribir este documento se encuentran las siguientes versiones en la página oficial:

- **Raspbian Buster with desktop and recommended software.** Imagen con escritorio y software recomendado basado en Debian Buster. Esta versión se lanzó en septiembre 2019, la versión del Kernel (núcleo del sistema operativo) es 4.19. (Tamaño de la imagen: 1,9 GB comprimida)
- **Raspbian Buster with desktop.** Imagen con escritorio basada en Debian Buster. Esta versión se lanzó en septiembre 2019, la versión del Kernel (núcleo del sistema operativo) es 4.19. (Tamaño de la imagen: 1,1 GB comprimida)
- **Raspbian Buster Lite.** Imagen mínima basada en Debian Buster. Esta versión se lanzó en septiembre 2019, la versión del Kernel (núcleo del sistema operativo) es 4.19. (Tamaño de la imagen: 426 MB comprimida).

Como se está diseñando un sistema Headless, utilizaremos la versión Lite, ya que pesa menos y utiliza los recursos mínimos y necesarios para poder utilizar la RPi.

En el segundo bloque del proyecto se explica paso a paso cómo instalar el sistema operativo Raspbian Buster Lite en la Raspberry Pi.

Conexión a la red

Existen muchas maneras para conectar la RPi a la red. En el proyecto se optan por dos opciones en función de las necesidades del proyecto:

- **Cable Ethernet:** esto consiste en conectar con el cable Ethernet al Router de nuestra casa igual que lo hacemos con nuestro ordenador de sobre mesa.
- **WiFi:** esto consiste en conectarnos a Internet igual que lo haríamos con nuestro ordenador portátil. Basta con tener el nombre de la Red y la contraseña de acceso.

La opción con WiFi es la preferente y la que se utiliza más a menudo durante la prueba de concepto y porque encaja mucho mejor con el concepto de internet de las cosas. En la segunda parte de este documento se explica cómo realizar estas conexiones.

Comunicación con la RPi

Una vez se conoce lo básico sobre la conexión a red es necesario saber cómo haremos para comunicarnos con la RPi. Como siempre existen muchas opciones para conseguir esto, pero la mejor opción y la que nos da un acceso completo a internet es por medio de la conexión segura con SSH (Secure Shell).

Secure Shell se emplea para establecer conexiones cifradas seguras entre dispositivos en la red. Podemos utilizar un cliente de terminal SSH para conectar con un servidor SSH a través del puerto 22 del RPi. Todo esto nos permite:

- Conectar en remoto con la RPi y ejecutar comandos.
- Transferir archivos desde y hacia el RPi usando protocolo SFTP (Protocolo de transferencia de archivos SSH).
- Redireccionar conexiones X11, lo que permite la computación en redes virtuales.

Existen muchos programas e incluso extensiones de navegador que permiten realizar conexiones SSH. El presente documento se utilizan PuTTY y Secure Shell App (extensión del navegador Chrome) para la realizar las conexiones.

Para la transferencia de archivos se utiliza Filezilla ya que se considera mucho más cómodo realizar la transferencia de archivos de este modo.

Comandos básicos para el control y configuración de la RPi

A continuación, se listan cada uno de los comandos básicos para que podamos interactuar con la RPi [3].

Tabla 5. Comandos básicos Linux 1. Fuente: [3]

Comandos básicos del sistema de archivos		
Nombre	Comando	Opciones y más información
Listar archivos	ls	- a los muestra todos, incluido los ocultos. - l muestra el formato largo. - R listado recursivo - r listado inverso. - t ordenar por última modificación. - S ordena por tamaños. - h muestra el tamaño en formato legible
Directorio actual	pwd	Indica el directorio actual. - P muestra la ubicación física
Cambiar directorio	cd	Cambia el directorio. cd e Intro o cd ~/ nos lleva al directorio raíz del usuario. cd / nos lleva al directorio raíz del sistema. cd .. nos lleva a un nivel superior en el árbol de directorios.
Crear un directorio	mkdir	Crea un directorio (una carpeta)
Borrar un archivo o directorio	rm	Borrar un archivo. - r borrado recursivo (se usa con directorios). - d borra directorios vacíos.
Borrar un archivo o directorio	cp	- r copia recursiva. - u copia solo si el origen es más reciente que el destino, o bien si no se encuentra el destino. - v (verbose) muestra la salida durante la copia.
Borrar un archivo o directorio	mv	- i solicita confirmación antes de sobrescribir. No hay opción - r para directorios. Mover al mismo directorio cambia su nombre.
Crear un archivo vacío	touch	Crea un archivo vacío o actualiza la fecha de modificación de un archivo existente.
Ver el contenido de un archivo.	more	Muestra el contenido de un archivo. Usamos la barra espaciadora para pasar a la siguiente página.
Obtener calendario	cal	Muestra un calendario basado en caracteres.
Comando habituales del gestor de sistemas y servicios systemd		
Comando		Descripción
systemctl		Lista todos los servicios en ejecución
Systemctl start ntp		Inicia un servicio. No persistente después de reiniciar.
Systemctl enable ntp		Habilita un servicio durante el arranque.
Systemctl restart ntp		Reinicia un servicio.
Comandos para transferir archivos entre carpetas		
sudo chmod 777 Nombre.txt		Permite transferir archivos con FileZilla (software)

Tabla 7. Comandos básicos Linux 2

Estructura de directorios de nivel superior en Linux	
DIRECTORIO	DESCRIPCIÓN
bin	Contiene los binarios ejecutables utilizados por todos los usuarios y está en la variable de entorno PATH de forma predeterminada. Otro directorio, /usr/bin , contiene ejecutable que no resultan básicos para el arranque o reparación del sistema.
boot	Contiene los archivos para arrancar el RPi
boot .back	Contiene una copia de /boot generada tras una actualización sistema.
dev	Contiene los nodos de dispositivo, enlazados a sus controladores.
etc	Archivos de configuración para el sistema local.
home	Contiene los directorios raíz (home) de las cuentas de usuario (/home/pi es el directorio raíz de la cuenta de usuario pi).
lib	Contiene las librerías estándar del sistema.
lost+found	Aquí figura los archivos sin vínculos que aparezcan tras ejecutar fsck (<i>file system check and repair</i> , comprobación y reparación del sistema de archivos). El comando mklost+found vuelve a crear el directorio lost+found si se ha eliminado.
media	Utilizado para montar dispositivos removibles , como tarjetas micro-SD.
mnt	Su uso típico es para montar sistemas de archivos temporales.
opt	Un buen lugar para instalar software opcional de terceros.
proc	Un archivo virtual que representa los procesos en ejecución en el RPi. Por ejemplo, si pasamos al directorio cd /proc y escribimos cat iomem , veremos algunos mapeos de asignaciones de memoria.
root	El directorio raíz de la cuenta root (superusuario) en las distribuciones Raspbian y Debian Linux. En muchas otras distribuciones, es el directorio /home/root .
run	Proporciona información acerca de la ejecución del sistema desde la última vez que arrancó.
sbin	Contiene ejecutables para la gestión del root.
srv	Almacena datos relacionados con ftp, servidores web, rsync, etc.
sys	Contiene un sistema de archivos virtual que describe el sistema.
tmp	Contiene archivos temporales.
usr	Contiene programas para todos los usuarios, junto con muchos subdirectorios como: /usr/include (archivos de cabecera C/C++), /usr/lib (librerías C/C++), /usr/src (código fuente del núcleo de Linux), /usr/bin (ejecutables del usuario), /usr/local (similar a /usr, pero para usuarios locales) y /usr/share (archivos compartidos entre los usuarios).
var	Contiene archivos variables como los log del sistema.

Comandos Git para el control de versiones**Tabla 8.** Comandos básicos Git (Gestión de repositorios)

OPERACIÓN	DESCRIPCIÓN	OPERACIÓN	DESCRIPCIÓN
git clone	Realiza la clonación desde el repositorio remoto.	git rm	Elimina un archivo o directorio del área de preparación.
git init	Crea un repositorio completamente nuevo.	git mv	Mueve o renombra un archivo o directorio en el área de preparación.
git pull	Fusiona cambios desde un repositorio maestro.	git log	Muestra un log (registro) con los commits. El histórico del proyecto.
git fetch	Encuentra los cambios en el repositorio maestro sin realizar la fusión.	git tag	Da nombre a un commit (por ejemplo: versión 2).
git status	Muestra el estado del proyecto.	git merge [name]	Fusiona la rama.
git add	Añade un nuevo archivo o edita un archivo existente.	git show	Obtiene detalles sobre el commit actual o sobre cualquier otro.
git diff	Muestra las diferencias que pasarán al repositorio cuando se haga commit.	git branch [name]	Crea una nueva rama de desarrollo. (Use el parámetro -d para borrar (delete)).
git commit	Realiza commit en el repositorio.	git checkout [name]	Cambia a otra rama de desarrollo.
git push	Envía los cambios desde un repositorio local a uno remoto.		

2.3.3. Recopilación de la capa de detección y recolección de datos.

Como podemos leer en los apartados anteriores para entender esta primera capa según el tipo de solución a ofrecer serán necesario disponer de sensores, actuadores, controladores y un dispositivo que haga de pasarela de comunicación IoT. Para la pasarela de comunicación se ha optado por utilizar la Raspberry Pi debido al bajo coste y porque son adecuados para realizar prototipos y/o pruebas de concepto (PoF).

Hemos visto que es necesario entender los fundamentos básicos de lo que es la RPi en sí mismo y también conocer los fundamentos básicos del funcionamiento del sistema operativo Linux (Raspbian Buster Lite).

A modo ilustrativo se muestra un diagrama de los componentes de esta primera capa.

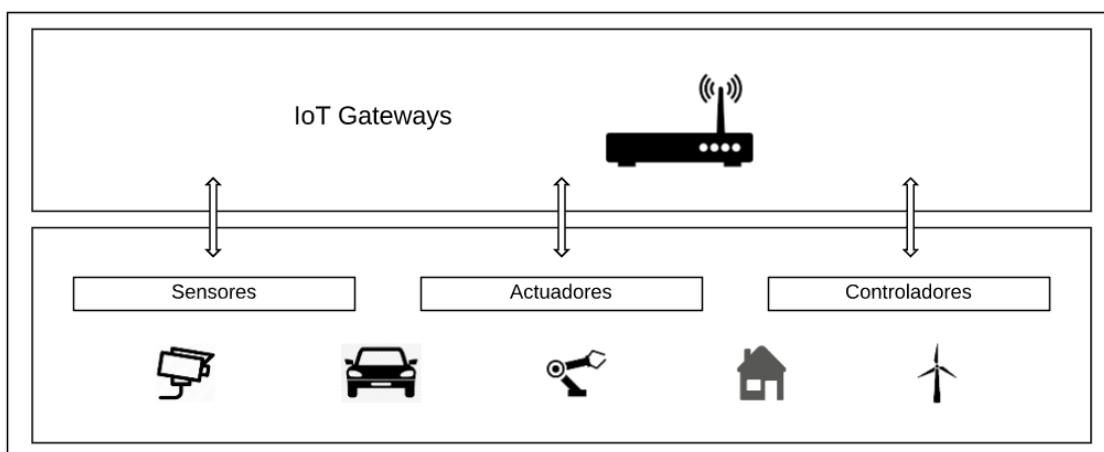


Figura 2.9. Capa de detección y recolección de datos. Fuente: Elaboración propia

3. Capa de transmisión de datos.

Para dar vida al concepto de IoT, es necesario la conexión, ya que sin ella resulta imposible el envío de los datos a otros dispositivos o los centros de almacenamiento para su posterior visualización, tratamiento y/o toma de decisiones. La conexión no tiene que ser necesariamente a Internet, la red de redes, también podemos dar vida al concepto de IoT en un sistema cerrado cumpliendo todas y cada una de las capas.

Para dar a conocer la gran diversidad de tecnologías de la comunicación que existen, es necesario primero hacer una clasificación de las redes en función de la distancia a la cual se pueden llegar a transmitir la información.

La clasificación de las redes en función de su distancia es la siguiente:

- **PAN (Personal Area Network)**, red de área personal traducido al castellano. En este grupo entra las redes formadas por dispositivos alejados a pocos metros entre ellos. Un ejemplo, sería unos auriculares con Bluetooth. Si se hablan de wearables, se utiliza el término **BAN (Body Area Network)**.
- **LAN (Local Area Network)**, red de área local. En este grupo entran las redes de dispositivos que engloban zonas que cubren un domicilio particular, una pequeña oficina o un edificio.
- **MAN o CAN (Metropolitan Area Network o Campus Area Network)**, red de área metropolitana o red de área de Campus. En este grupo se encuentran redes capaces de conectar edificios y barrios de ciudades entre ellos. Un ejemplo de este tipo de red, es la red de una universidad.
- **WAN (Wide Area Network)**, red de área amplia. Se trata de redes capaces de conectar grandes ciudades y países entre ellos.

A esta clasificación no hay que olvidarse de las tecnologías de conectividad inalámbrica, las cuales son igual que las anteriores, pero sin cables y para referirnos a ellas se pone una W de Wireless delante. Ejemplos WPAN, WLAN, WMAN, etc.

Las tecnologías de conectividad inalámbrica suelen ser las más adecuadas para el concepto de IoT, ya que suelen haber lugares en los cuales resulta difícil acceder con cables. En la siguiente imagen (**Figura 3.1**), se da a conocer algunas de las tecnologías de la comunicación que se están utilizando en la actualidad relacionadas con internet de las cosas. A todo esto, también hay que decir que no existen estándares ni protocolos establecidos de forma definitiva y se están creando muchos más para dar respuesta a la gran cantidad de dispositivos que se prevé que estén conectados en un futuro no muy lejano.

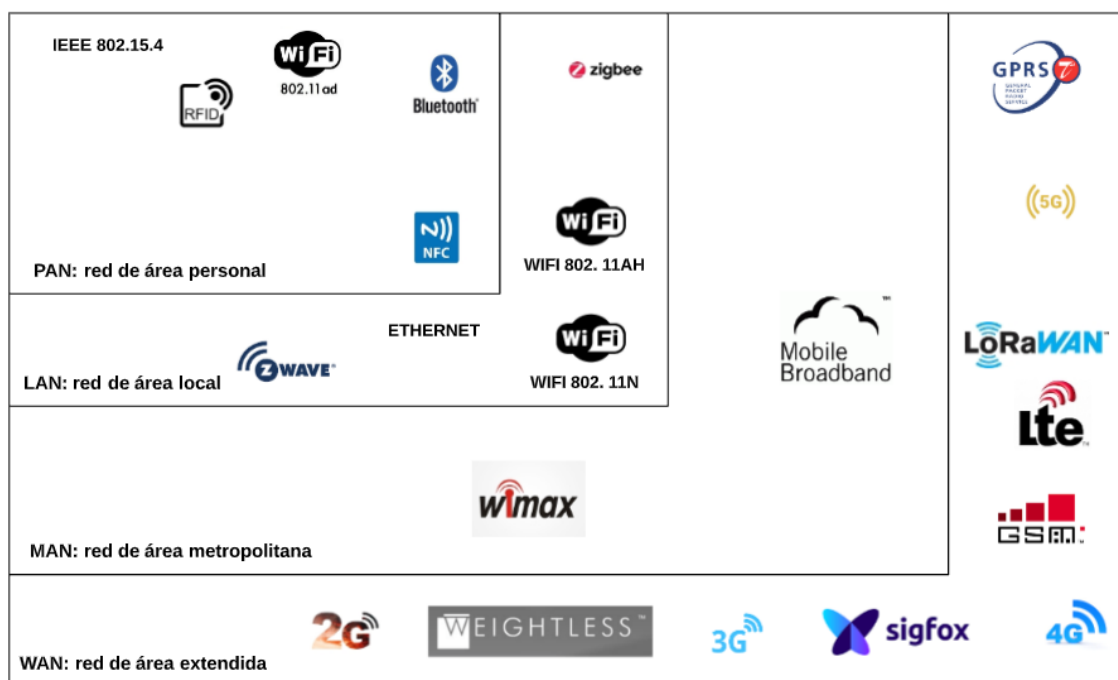


Figura 3.1. Tecnologías de comunicación para la transmisión de datos. Fuente: Elaboración propia

Ejemplos de tecnologías en fase experimental para el concepto de IoT son en el momento de redactar este documento son, **Thread** (protocolo moderno que usa IPv6), **DECT** (Digital Enhanced Cordless), **ULE** (Ultra Low Energy), **NB-IoT**, **LTE Cat 1**, **LTE CatM1** y muchos más.

No es conveniente entrar en detalle en cada una de estas tecnologías ya que se sale del alcance del presente documento. Es suficiente con tener una idea global de las diversas tecnologías que existen y estudiar cada una por separado para poder elegir mejor que tecnologías es adecuada para nuestra solución IoT.

Otro concepto que viene bien tener claro es el Modelo OSI y el Modelo TCP/IP, ya que son los fundamentos para entender cómo está ocurriendo la transmisión de datos, ya sea por ondas electromagnéticas o por cables.

Modelo OSI

Modelo de interconexión de sistemas abiertos o como comúnmente se conoce, modelo OSI (Open System Interconexión). Se trata de un modelo de referencia para los protocolos de la red.

Una normativa formada por siete capas que define las diferentes fases por las que deben pasar los datos para viajar de un dispositivo a otro sobre una red de comunicaciones [4].

En la **Tabla 9** podemos ver los distintos protocolos que se utilizan en cada capa. Muchos de estos protocolos los utilizaremos indirectamente en nuestro proyecto y sin darnos cuenta. No se entra en más detalle sobre este tema ya que no aportaría valor y tampoco es necesario para poder llevar a cabo el proyecto con eficacia lo único importante que se explica y que es importante conocer es el protocolo MQTT de la capa de aplicación la capa 7.

Tabla 9. Ejemplo de protocolos utilizados por capa OSI. Fuente: [wikipedia](https://es.wikipedia.org/wiki/Modelo_de_capas_de OSI)

7. CAPA DE APLICACIÓN	FTP, HTTP, MQTT , POP, SSH, XMPP, TELENET, etc.
6. CAPA DE PRESENTACIÓN	Presenta la información.
5. CAPA DE SESIÓN	Protocolo de la capa de sesión: - Protocolo RPC (llamada a procedimiento remoto). - SCP (Secure Copy). - ASP (Protocolo de sesión APPLE TALK).
4. CAPA DE TRANSPORTE	Ejemplo de protocolo de transporte de internet: - UDP (Protocolo de datagramas de usuario). - TCP (Protocolo de control de transmisión).
3. CAPA DE RED	IP (IPsec, IPv4, IPv6,), OSPF, IS-IS, ICMP, ICMPv6, IGMP
2. CAPA DE ENLACE DE DATOS	Direccionamiento físico (MAC y LLC)
1. CAPA FÍSICA	Ejemplo de protocolos: - IEEE 802.11x Wi-Fi capas físicas. - Bluetooth capa física. - USB capa física. - GSM interfaz radio.

Modelo TCP/IP

Misma función que el modelo OSI, solo que en vez de siete capas son 4 capas. Aquí podemos ver claramente que hay protocolos que nos resultan familiares en nuestro día a día como, HTTP, HTTPS, SSH, IPv4, Ethernet y Wifi. [5]

Tabla 10. Ejemplo de protocolos utilizados por capa TCP/IP. Fuente: [wikipedia](https://es.wikipedia.org/wiki/Modelo_de_capas_de TCP/IP)

4. CAPA DE APLICACIÓN	BGP, DHCP, DNS, FTP, HTTP, HTTPS, IMAP, LDAP, MGCP, MQTT , POP, IMAP, SSH, XMPP, TELNET, TLS/SSL, etc.
3. CAPA DE TRANSPORTE	TCP, UDP, DCCP, SCTP, RSVP, etc.
2. CAPA DE INTERNET	IP (IPv4 y IPv6), ICMP, ICMPv6, ECN, IGMP, IPsec, etc.
1. CAPA DE ACCESO A LA RED	ARP, NDP, OSPF, Tunnels (L2TP), PPP, MAC (Ethernet, Wi-Fi, DSL, ISDN, FDDI), etc.

Relación del modelo OSI y el modelo TCP/IP. Aquí se muestra una comparación de los dos modelos.

Tabla 11. Relación de los dos modelos. OSI vs TCP/IP. Fuente: [wikipedia](https://es.wikipedia.org/wiki/Modelo_de_capas_de OSI)

MODELO OSI	MODELO TCP/IP
7. CAPA DE APLICACIÓN	4. CAPA DE APLICACIÓN
6. PRESENTACIÓN	
5. SESIÓN	
4. TRANSPORTE	3. CAPA DE TRANSPORTE
3. RED	2. CAPA DE INTERNET
2. ENLACE DE DATOS	1. CAPA DE ACCESO A LA RED
1. FÍSICA	

Para el proyecto que del presente documento no interesa conocer al detalle cada una de estas tecnologías, pero es importante conocer las opciones que existen, para poder estudiarlas y elegir la mejor tecnología que nos convenga en función de la solución IoT a ofrecer.

De todas las tecnologías mencionadas anteriormente nos interesa el Wifi, Ethernet, MQTT y los protocolos de comunicación de internet (HTTP, MQTT, etc)

A la hora de ofrecer una solución IoT es importante elegir bien la tecnología que asegure una buena transmisión de los datos.

MQTT (Message Queue Telemetry Transport).

MQTT (Message Queue Telemetry Transport), como hemos visto anteriormente forma parte de la capa de aplicación del modelo OSI (capa 7) y el modelo TCP/IP (capa 4), por lo que estamos hablando de un protocolo de comunicación. En este caso se trata de un protocolo de red de publicación/suscripción que transporta mensajes entre dispositivos. Al ser ligero, requerir poco ancho de banda, alta latencia, poco consumo y permite la comunicación entre dispositivos [6]. Esto ha permitido que sea un protocolo actualmente muy adecuado para el mundo del IoT.

Utilizaremos este tipo de protocolo para nuestro proyecto, en vez del modelo petición/respuesta que se utiliza en el protocolo HTTP.

El protocolo MQTT funciona de la siguiente manera:

Broker: servidor central encargado de todas las conexiones entre clientes y el almacenamiento de todos los mensajes.

Cliente: cualquier dispositivo que se conecta al bróker y puede publicar y recibir mensajes.

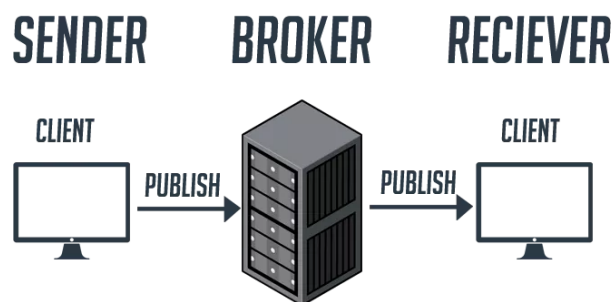


Figura 3.2. Actores en el protocolo MQTT. Fuente: [mntolia](http://mntolia.com)

Los clientes o dispositivos que están conectados al bróker pueden comunicarse entre ellos a través de tópicos. Un tópico se puede definir como un canal de radio o televisión en el cual cualquier persona que se conecte a ese canal puede recibir la información emitida por ese canal. Un tópico son palabras como, por ejemplo, HumedadSensor, Luzled, sensor/temperatura. Palabras creadas por el programador, que hacen referencia a la información del tópico.

Publicar, en el protocolo MQTT se refiere a emitir información o datos. Y suscribirse significa sintonizar con el canal que publica un determinado tópico.

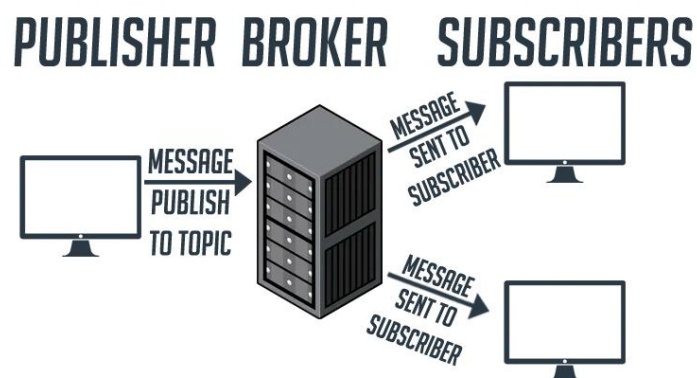


Figura 3.3. Diferentes subscriptores conectados a un mismo tópico o canal. Fuente: [mntolia](http://mntolia.com)

En la segunda parte del presente documento se muestra cómo se implementa este protocolo de comunicación.

Siguiendo con el símil de las capas, se muestra a modo ilustrativo como se compone el concepto de IoT con las capas definidas hasta ahora.

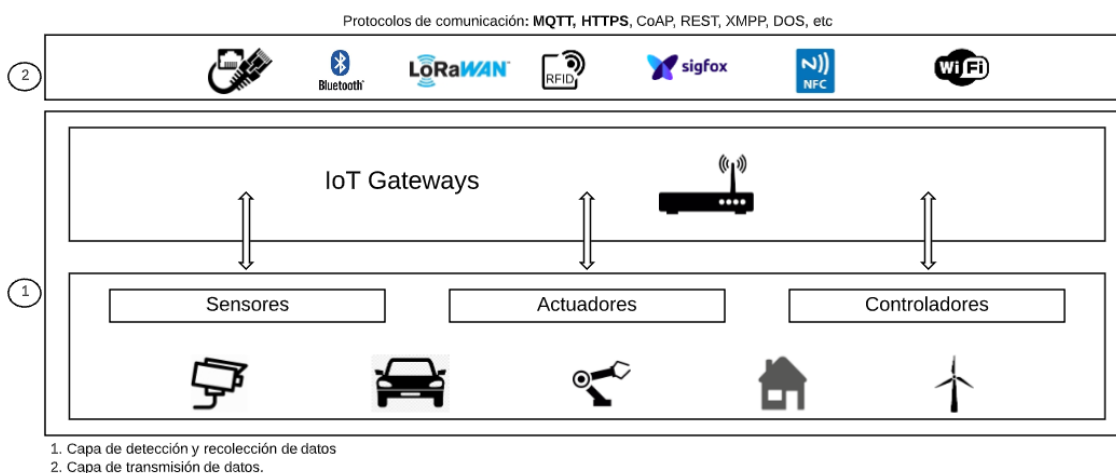


Figura 3.4. Capa de transmisión y la capa de detección de datos juntas. Fuente: Elaboración propia

4. Capa de almacenamiento y capa de analítica de los datos. Cloud computing.

La capa de almacenamiento de datos (capa 4) y la capa de analítica de datos (capa 5) se explican en este apartado como si fuesen una única capa, ya que estas dos fases del concepto IoT ocurren en una misma plataforma. Más adelante se explican con mayor detalle.

4.1. Capa de almacenamiento de datos

La función de esta capa podría consistir simplemente en almacenar la información en una base de datos convencional (base de datos relacional) o una base de datos no relacional de la propia Raspberry Pi o de nuestro ordenador personal y ahí acabaría la historia. Si se hace esto no estaríamos dando vida al concepto de IoT, que son millones de dispositivos comunicándose entre sí, e incluso se habla de billones de dispositivos enviando datos a un bróker (servidor central) y este bróker a su vez enviado datos a los dispositivos que están suscritos a un determinado tópico. Por lo que estamos hablando de un ordenador muy potente (Cloud Computing) capaz de almacenar, procesar y analizar grandes cantidades de información (Big Data).

¿Qué es Big Data? Con lo descrito en el párrafo anterior podemos hacernos una idea del significado de Big Data. Big Data son grandes cantidades de datos, que pueden ser estructurados (base de datos relacional) como no estructurados (base de datos no relacional). Estos datos pueden ser de millones de dispositivos conectados o de cualquier otra fuente.

Ejemplos de Base de Datos Relacionales (SQL): MySQL, SQLite, PostgreSQL, etc.

Ejemplo de Base de datos No Relacional (NoSQL): DynamoDB, MongoDB, Redis, etc.

La gran cantidad de datos (Volumen), suele venir de una gran variedad de fuentes (Variabilidad) y con una cierta velocidad de crecimiento (Velocidad), difícil de ser gestionada por nuestros ordenadores convencionales y esto hace que sea necesario el Cloud Computing [7].

¿Qué es el Cloud Computing? Computación en la nube o simplemente La nube, consiste simplemente en ofrecer servicios a través de Internet. Estos servicios pueden ser tan simples como nuestro proveedor de correo electrónico hasta algo más complejo como tener un programa informático como AutoCAD listo para ser utilizado a través de nuestro navegador web.

Ahora que ya tenemos nos hacemos una idea, una definición formal, sería la siguiente:

Tecnología (centro de datos) capaz de proveer acceso remoto a softwares, almacenamientos de cualquier tipo de archivos y procesamiento de datos por medio de Internet ofreciendo una alternativa mucho más flexible, segura y escalable en comparación con nuestro ordenador personal. [8]

La nube puede ser privada, pública, híbrida o comunitaria. Para la prueba de concepto se utiliza una nube publica, ya que es mucho más barata en relación con las otras. Una nube pública, como su mismo nombre indica es aquella en la cual cualquier persona, entidad o empresa puede hacer uso de ella de manera simultánea, pero de manera separada. La empresa encarga de proveer la

nube se hace cargo de su mantenimiento y seguridad de tal manera que los usuarios simplemente se dedican a hacer uso de ella.

Los servicios en la nube se pueden ofrecer de distintas maneras:

- **SaaS (Software as a Service o Software como servicio).** Se trata de los programas informáticos que instalamos en nuestros ordenadores personales como Skype, AutoCAD, Office, etc, ahora son accesibles a través de nuestro navegador web y ya no es necesario que nuestro ordenador disponga de todos los recursos computacionales que requiere el programa informático.
- **PaaS (Platform as a Service o Plataforma como Service).** Se trata de un entorno en el cual el proveedor de la nube proporciona los recursos necesarios para el desarrollo e implementación de aplicaciones. Ejemplos de estas aplicaciones son Heroku, Force.com, Magento Commerce Cloud, etc.
- **IaaS (Infrastructure as a Service o Infraestructura como servicio).** Esta infraestructura proporciona la capacidad de proporcionar ciertos recursos, concretamente, servidor, almacenamiento, seguridad y el propio centro de datos. Y todo lo demás es gestionado por la empresa proveedora del servicio en la nube. Un ejemplo es, AWS EC2 o Google Compute Engine (GCE), en los que ambos permiten manejar una máquina virtual en la nube.

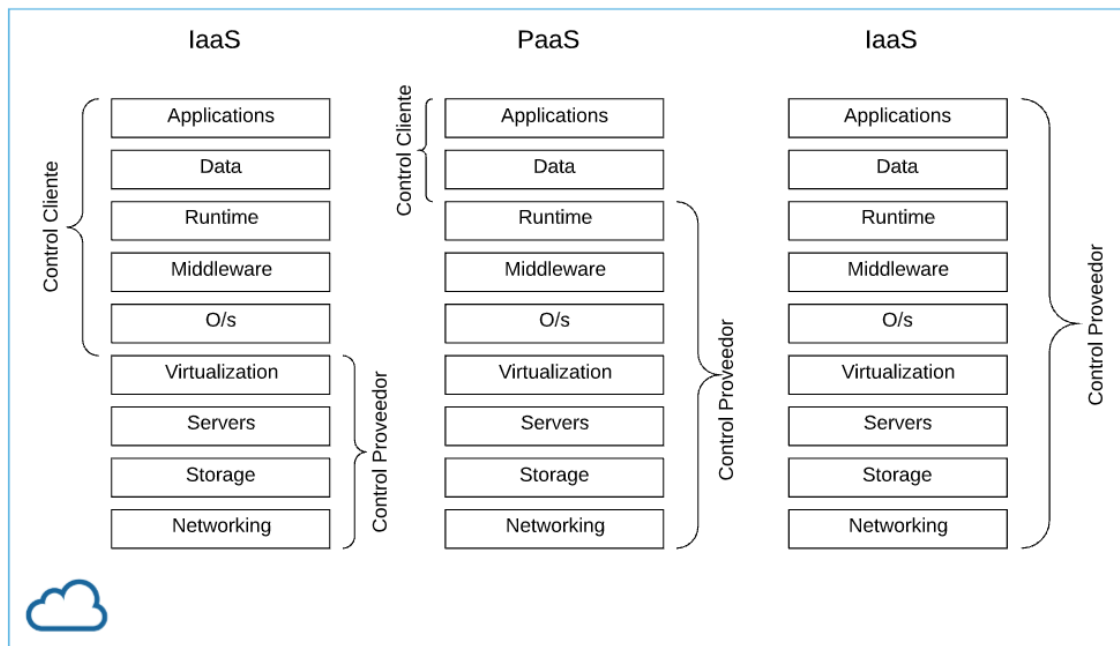


Figura 4.1. Estructura de los servicios en la nube. Fuente: Elaboración propia



Figura 4.2. Estructura de los servicios en la nube. Fuente: Microsoft Azure.

Ahora que entendemos conceptualmente el Cloud Computing o La nube, es necesario mencionar otros dos nuevos conceptos que tienen relación con la nube y el concepto de IoT. Estos son dispositivos Fog (Niebla) y dispositivos Edge (Borde). Estos dos dispositivos a pesar de ser muy diferentes tienen un solo objetivo: filtrar, analizar y reducir la cantidad de información a enviar a la nube para mejorar la eficiencia del sistema en global, intentando evitar cuellos de botella en el tráfico hacia esta.

Estos dispositivos se utilizan en soluciones para IoT en el sector Industrial (IIoT), por lo que aquí no lo tocaremos, es simplemente a modo de mención.

Estos dispositivos se utilizan cuando entran en juego millones de sensores, un volumen significativo de tráfico de datos y sistemas centrales alejados del lugar donde se generan. Este sería el caso de una ciudad inteligente o Smart City.

4.2. Capa de analítica de datos

Una vez se han almacenado los datos, viene la siguiente fase o capa del concepto de IoT, en la cual consiste en procesar los datos mediante tecnologías y herramientas de analítica de Big Data. Estas tecnologías y herramientas giran en todo a la minería de datos (Data Mining), el aprendizaje automático (Machine Learning) y el aprendizaje profundo (Deep Learning), las dos últimas herramientas son un subconjunto de la inteligencia artificial.

A continuación, se procede a definir estos nuevos conceptos para hacernos una idea del valor que a porta esta capa al concepto de IoT.

Data Minig (DM) o minería de datos, es un proceso que consiste en extraer utilidad y comprensión a la gran cantidad de datos que se han adquirido en distintos formatos.

Machine Learning (ML) o aprendizaje automático, rama de la inteligencia artificial que busca construir algoritmos que permiten a los computadores “aprender” a partir de conjuntos de datos y obtener como resultado un modelo que permita realizar procedimientos basándose en dichos datos y no en instrucciones estáticas. [1].

Deep Learning (DL) o aprendizaje profundo, subcategoría del aprendizaje automático, trata de redes neuronales para mejorar cosas tales como el reconocimiento de voz, la visión por computadora y el procesamiento del lenguaje natural. [1].

Como podemos ver son ramas del conocimiento que se alejan bastante del objetivo de la prueba de concepto a realizar en el segundo bloque, y no se realiza nada en relación a esta capa del concepto de IoT.

4.3. Amazon Web Services (AWS).

Existen muchas plataformas (PaaS) centradas en ofrecer servicios especializados de IoT. Estas plataformas simplemente ofrecen opciones prediseñadas por el proveedor del servicio sin ser posible poder modificar nada más allá de lo que te permita el servicio. Ejemplo de estas plataformas son ThinkSpeak o IBM Bluemix Internet of Things.

En el presente documento se construye un SaaS (Software As A Service), con la única funcionalidad de visualizar los datos. En el cual el cliente simplemente tenga que centrarse en utilizar la aplicación. Para hacer esto de todas las plataformas que existen y que ofrecen un control completo para el creador de la aplicación SaaS, se ha elegido AWS. Otras plataformas que cumplen la misma función son Google Cloud Platform (GCP) y Microsoft Azure.

Muchos de estos servicios tienen sus homólogos con las plataformas competidoras, por lo que no hay que verse limitado a hacer uso únicamente de Amazon Web Services (AWS). Lo mejor es utilizar la plataforma con la que uno sienta más cómodo, sepa utilizar y económicamente sea viable.

A continuación, se da a conocer Amazon Web Services y los servicios concretos que se utilizan para el proyecto que nos ocupa.

Amazon Web Services (AWS), plataforma de computación en la nube pública, que ofrece una gran variedad de servicios de computación, que van desde el almacenamiento en la nube hasta servicios de Machine Learning pasando por la contratación de máquinas virtuales, servicios de Internet de las cosas, herramientas para desarrolladores, etc.

A continuación, se dan a conocer los servicios más relevantes en relación al presente proyecto.

A día de la redacción todavía existe la capa gratuita, que son un conjunto de servicios que son gratuitos, pero con ciertas limitaciones de tiempo y almacenamiento. Nos limitaremos a utilizar los servicios de esta capa para no incurrir en grandes costes para la realización de la prueba de concepto.

Tabla 12. Algunos servicios de la nube de AWS. Fuente: elaboración propia

Nombre del Servicio	Pequeña descripción del servicio
Amazon EC2	Amazon Elástico Compute Cloud, es el nombre completo. En términos que se puedan entender es la contratación de una máquina virtual con los sistemas operativos disponibles por la plataforma.
Amazon S3	Amazon Simple Storage Service, es un servicio de almacenamiento de objetos, en otras palabras, es como tener un pendrive donde puedes almacenar cualquier tipo de archivo, solo que este pendrive es escalable y seguro.
Amazon DynamoDB	DynamoDB, es una base de datos no relacional, es decir una base de datos NoSQL o no estructurado.
AWS Lambda	Es una Función como servicio (FaaS), este servicio permite ejecutar código en Python, Node.js o Java sin aprovisionar ni

	administrar un servidor. Muy utilizado en aplicaciones Serverless.
Amazon SNS	Amazon Simple Notification Service (SNS), es un servicio de mensajería de publicación/suscripción completamente administrada, de alta disponibilidad, seguro y con durabilidad que permite desacoplar microservicios, sistemas distribuidos y aplicaciones sin servidor. [Definición de la página web oficial]
Amazon CloudWatch	Amazon CloudWatch, es un servicio de monitorización y observación creado para ingenieros de DevOps, desarrolladores, ingenieros de fiabilidad de sitio (SRE) y administradores de TI. [Definición de la página web oficial]
Amazon API Gateway	Amazon API Gateway, es un servicio completamente administrado que facilita a los desarrolladores la creación, la publicación, el mantenimiento, el monitoreo y la protección de API a cualquier escala. [Definición de la página web oficial]
Amazon Cognito	Amazon Cognito le permite, de manera rápida y sencilla, incorporar a sus aplicaciones web y dispositivos móviles funcionalidades como el control de acceso, la inscripción y el inicio de sesión de los usuarios. [Definición de la página web oficial]
Amazon QuickSight	Amazon QuickSight, es un servicio de inteligencia empresarial rápido y basado en la nube que permite proporcionar información a todos los integrantes de su organización fácilmente. [Definición de la página web oficial]
AWS IoT Analytics	AWS IoT Analytics, es un servicio completamente administrado que facilita la ejecución e instrumentación de análisis sofisticados de enormes volúmenes de datos de IoT, sin tener que preocuparse por todo el costo y la complejidad normalmente obligatorios en la creación de una plataforma de análisis de IoT. [Definición de la página web oficial]
Amazon SQS	Amazon Simple Queue Service (SQS) es un servicio de colas de mensajes completamente administrado que permite desacoplar y ajustar la escala de microservicios, sistemas distribuidos y aplicaciones sin servidor. [Definición de la página web oficial]
AWS IoT	Conjunto de servicios con la finalidad de ofrecer soluciones para Internet de las Cosas.
Amazon Route53	Servicio DNS (Domain Name System). En otras palabras, servicio que sirve para comprar un dominio y redireccionar nuestro servidor web o nuestra aplicación sin servidor (serverless), con un nombre fácil y corto de recordar.
AWS Cloud9	AWS Cloud9, es un entorno de desarrollo integrado (IDE) basado en la nube que le permite escribir, ejecutar y depurar su código solo con un navegador. [Definición de la página web oficial]
IAM	Sirve para controlar de forma segura nuestra cuenta de AWS, permite dar acceso a terceros, limitar o ampliar los permisos a la utilización de diversos servicios de nuestra cuenta.

Existen muchísimos más servicios en la plataforma de AWS, pero con conocer estos ya será suficiente para poder crear la prueba de concepto del presente documento.

4.4. Resumen Capa de almacenamiento y capa de analítica de los datos

Para finalizar y continuando con el símil de las capas, se muestra a modo ilustrativo, cómo se va completando el diagrama con los conocimientos que ya sabemos hasta ahora.

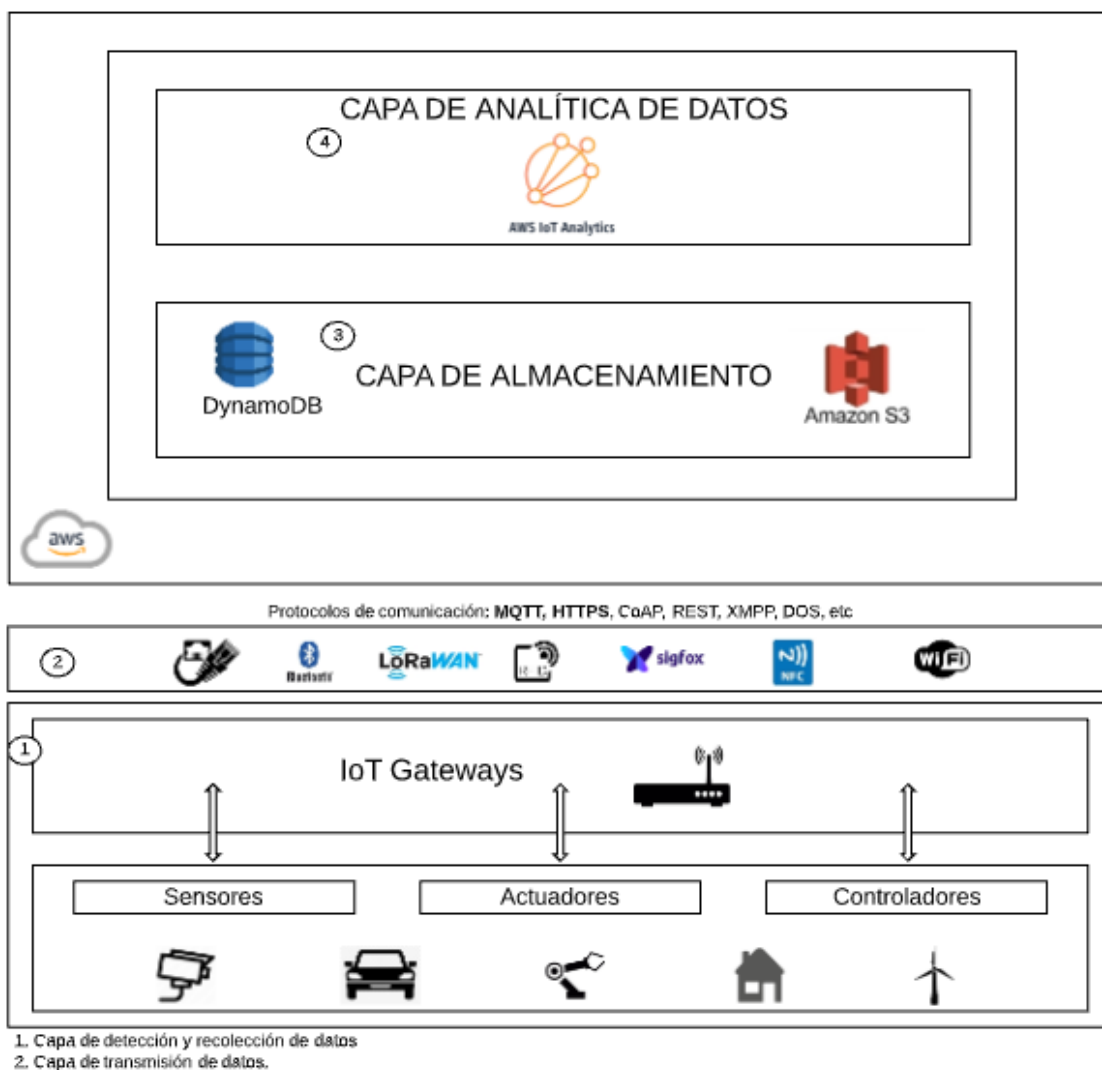


Figura 4.3. Las cuatro primeras capas del concepto de IoT Fuente: Elaboración propia

Como podemos ver el concepto de IoT se va haciendo cada vez más complejo a medida que van involucrando más capas.

5. Capa de presentación y visualización de los datos.

La capa de presentación y/o visualización de los datos consiste simplemente en una página web, una aplicación móvil o un aplicativo de escritorio, el cual te permite ver y realizar ciertas funciones programadas por el desarrollador de la aplicación. Estas funciones pueden ser la visualización de los datos en tiempo real, la visualización de los datos de uno a otro periodo del tiempo, descargar los datos en cualquier formato por medio de reportes, encendido o apagado de un determinado activo, control del acceso al aplicativo, analítica de los datos, avisos de cuando se alcanza un determinado valor por medio del móvil o el e-mail, etc. Todo depende de las habilidades del programador.

En la Figura 5.1, se puede ver el conjunto de las 5 capas del concepto de IoT. Este apartado no tiene mucha más vuelta de hoja, por lo que pasaremos a explicar los lenguajes de programación utilizados para poder crear la capa de presentación.

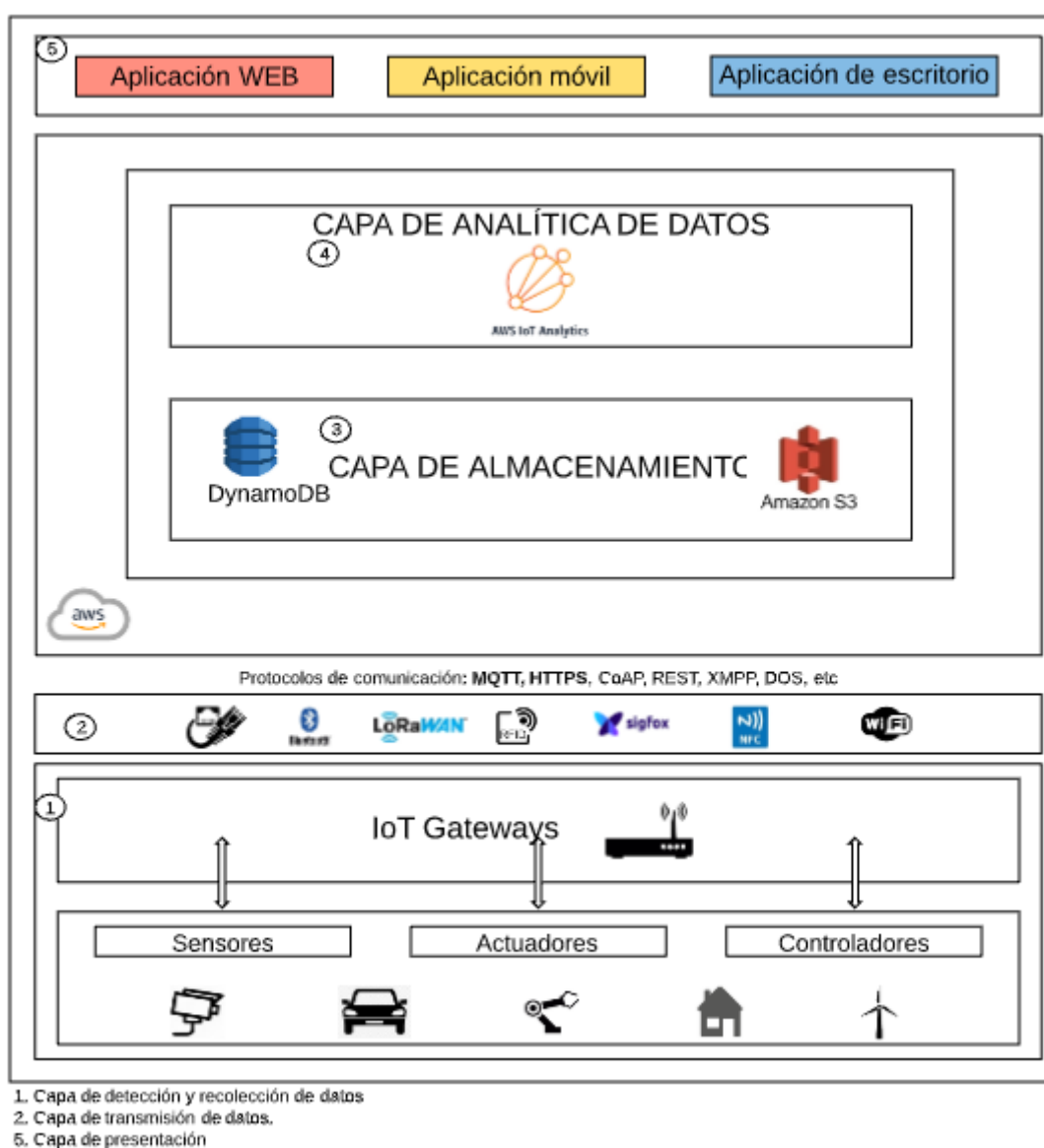


Figura 5.1. Las cinco primeras capas del concepto de IoT. Fuente: Elaboración propia

5.1. Lenguajes de programación utilizados.

Para poder ofrecer una solución IoT con una capa de presentación creado por uno mismo se han utilizado los siguientes lenguajes informáticos:

- Para la parte de Frontend, aquella parte de una aplicación que se centra en lo que ve el usuario, se utiliza:
 - **HTML (HyperText Markup Language)**, lenguaje que sirve para indicar cómo va ordenado el contenido de la página web.
 - **CSS (Cascading Style Sheets)**, lenguaje de diseño gráfico, es decir añaden contenido gráfico al contenido en HTML.
 - **Bootstrap**, *es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. [Definición wikipedia]*
 - **JavaScript**, *lenguaje de programación orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. [Definición wikipedia]*
- Para la parte Backend, aquella que se centra en el lado del servidor:
 - **Python**, *se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. [Definición wikipedia]*
 - **DynamoDB**, otra base de datos noSQL creada por AWS, igual que MongoDB.

6. Seguridad y protección en el concepto de IoT.

La seguridad no es un tema baladí a la hora de crear un proyecto de IoT, probablemente sea la fase más crítica de todo el proceso ya que si no se tiene en cuenta podemos ser muy vulnerables frente a terceros (personal no autorizado, hackers, etc).

El concepto de IoT se está utilizando en muchas industrias, como por ejemplo el transporte, la salud, logística y distribución, monitorización para la analítica predictiva y preventiva de grandes activos (plantas fotovoltaicas.). No tener en cuenta el aspecto de la seguridad en todas las capas puede producir grandes catástrofes.

Existe una rama de la informática conocida como ciberseguridad, no se entra en este tema ya que se aleja del objeto del presente proyecto. En este apartado se limite a describir de manera global como se emprenden algunas acciones de seguridad para la prueba de concepto, en las capas más susceptibles de ser atacadas.

Protección de la capa de detección y adquisición de los datos.

En esta capa, tenemos el dispositivo IoT Gateway, que en nuestro caso se trata de una Raspberry Pi Modelo B+.

Una manera simple y fácil de proteger la RPi, frente acceso a terceros es cambiando la contraseña que viene por defecto para acceder al sistema operativo.

Otra manera es comunicarse con la RPi por medio de SSH (o Secure SHell) con el software PuTTY o la App de Chrome para conexiones remotas

Protección de la capa de transmisión de los datos.

No se ha aplicado ningún método de seguridad en esta capa. No porque no sea necesario, sino que se desconocer cómo podrían hackearnos los datos enviados a la nube a través de esta capa.

Protección en la nube.

Esto abarca las dos capas la de almacenamiento y la capa analítica de los datos. Al utilizar una plataforma para realizar esta fase, la propia plataforma ya proporciona unos métodos que si se siguen correctamente ya te protegen del acceso frente a terceros. En el segundo bloque se podrá ver paso a paso cómo se van adoptando estas buenas prácticas.

Generación de claves y certificados para crear conexiones SSL seguras entre clientes y servidores de MQTT a través del servicio de AWS IoT es uno del método utilizados en esta capa.

AWS IAM, es otro servicio de la nube que proporciona credenciales de acceso que limitan o aplian el acceso de tu cuenta de AWS.

En el segundo bloque se implementan todas estas buenas prácticas.

Protección en la capa de presentación.

Es posible que, por medio de la capa de presentación, si diseñamos una página web, cualquier desarrollador con unos mínimos conocimientos pueda acceder al código de la parte Frontend. Por lo que debemos ser lo suficientemente prudentes a la hora de desarrollar la aplicación de no poner contraseñas de acceso a nuestros datos.

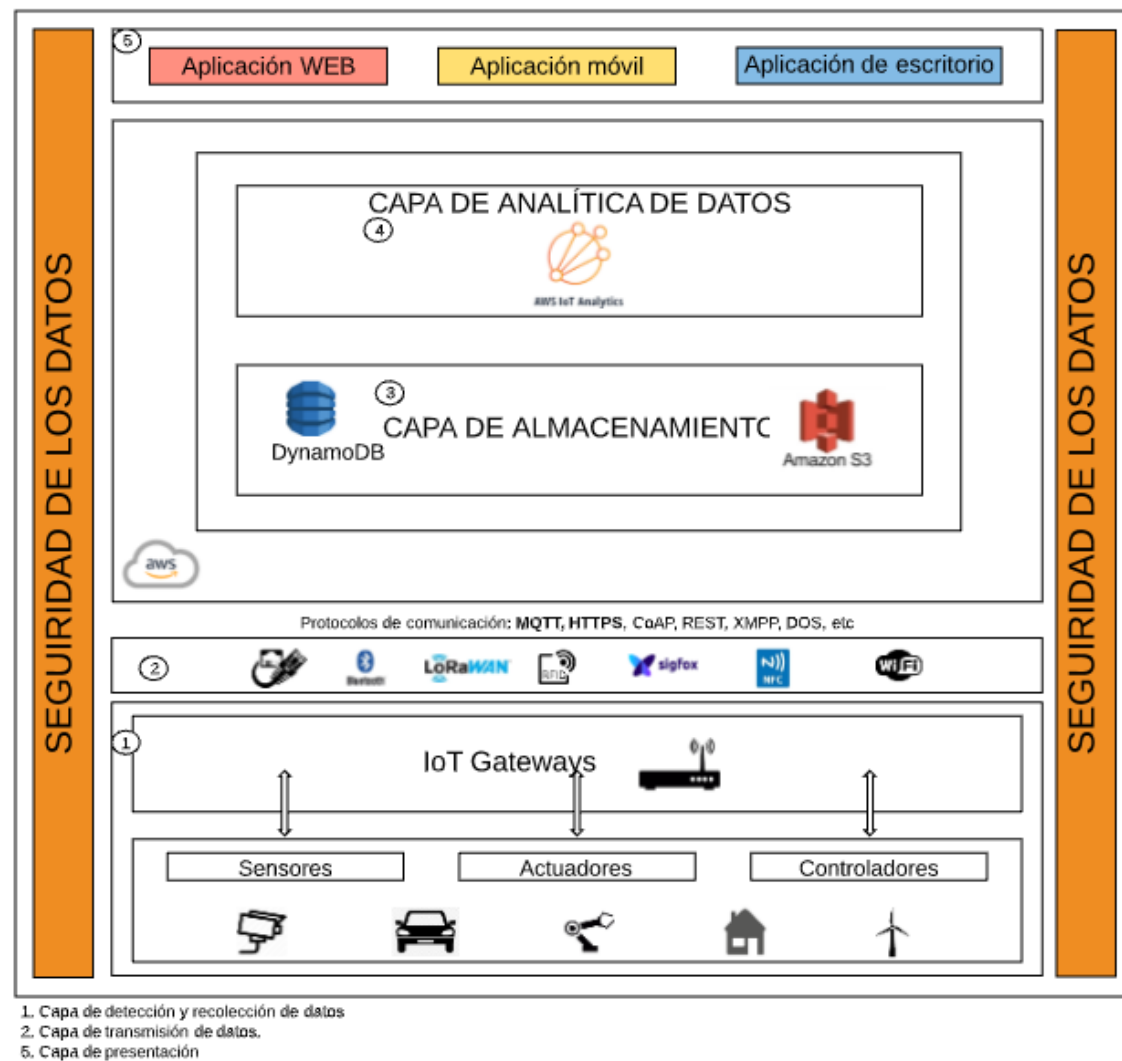


Figura 6.1. Todas las capas que hace posible y segura el concepto de IoT. Fuente: Elaboración propia

7. Aplicaciones IoT en el sector energético.

Llegados a este punto ya sabemos que Internet de las Cosas va a cambiar la forma en que funcionan muchas industrias, una de ellas es la industria energética. Si buscamos por internet nos encontraremos con términos como Energía 4.0, Smart Cities, Smart Grids, Industria 4.0 o Cuarta revolución industrial, todas hacen referencia a lo mismo, el Internet de las Cosas como espina dorsal para promover la transformación digital de todos los sectores, en este caso, la energía.

El concepto de IoT en el sector energético está permitiendo los siguientes cambios en este sector:

- Mejor monitoreo y gestión remota de los activos.
- Una red eléctrica mejor distribuida y mejor gestionada (Smart grid).
- Clientes más informados, ya que pueden acceder a la visualización de sus datos de consumo.
- Contribución a un mayor ahorro y eficiencia de los edificios.
- Mejor mantenimiento de las infraestructuras energéticas, gracias a la analítica preventiva y predictiva.
- Y un sinfín de mejoras en el sector energético.

A continuación, vamos a dar conocer ejemplo de cómo se está aplicando el concepto de IoT en el sector de la energía.

7.1. Schneider Electric y Microsoft Azure. IoT para la energía.

En la página web oficial de Microsoft Azure (proveedor de servicios en la nube), dan a conocer cómo la empresa Schneider Electric (Empresa del sector energético) ha utilizado los servicios de la nube de Microsoft Azure para crear un panel informativo global de placas solares en Nigeria.

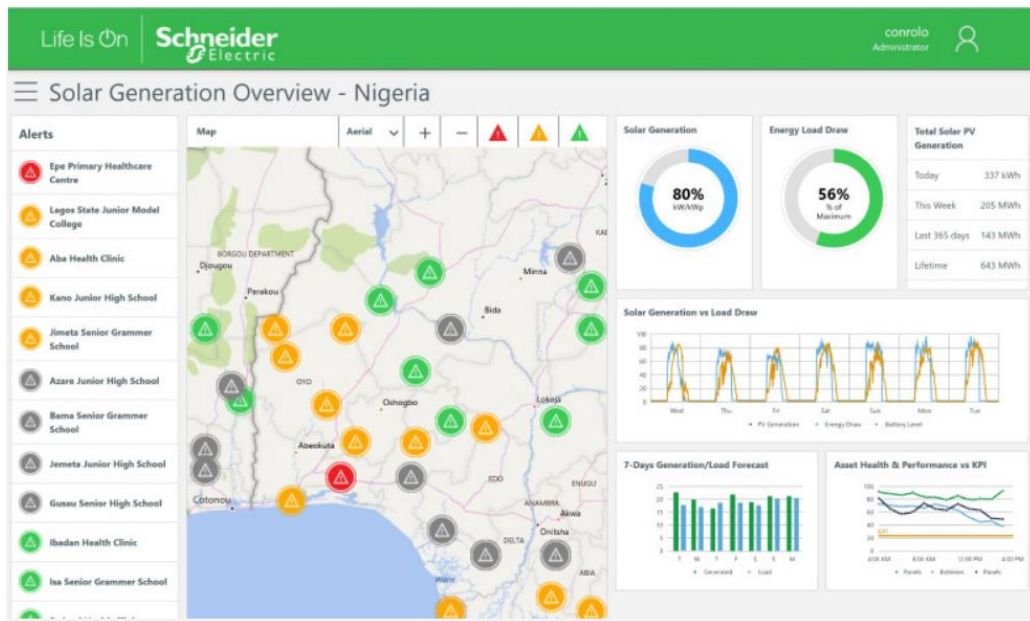


Figura 7.1. Panel informativo Schneider Electric y Microsoft Azure. Fuente: Schneider Electrico

Este panel de control ofrece:

- Administración de forma remota de todas las instalaciones.
- Se puede ver la ubicación de cada uno de los activos.
- Supervisión del estado y el rendimiento de todas las instalaciones.
- Dispone de servicio de alertas en caso de que haya problemas en las baterías o paneles solares.
- Permite enviar las alertas a los operadores de la planta para que procedan a su corrección.

Todo esto y mucho más permite el panel de control creado por Eschneider electric y Microsoft Azure.

Fuente: <https://azure.microsoft.com/es-es/overview/iot/industry/energy/>

7.2. Sentilo Terrassa y Sentilo BCN.

Tanto la ciudad de Terrassa como la de Barcelona están dando vida al concepto de ciudad inteligente, aplicando el concepto de IoT. Se ha desplegado una red de sensores por estas dos ciudades y se ha creado una plataforma en la cual se pueden visualizar los datos capturados por estos sensores.

Los sensores desplegados son los siguientes:

- Temperatura y humedad
- Analizador de redes
- Medidores de sonido ambiental
- Anemómetros
- Caudalímetros
- Contadores eléctricos
- Sensores para la ocupación de las plazas de parking
- Medidor del flujo de personas
- Sensor de meteorología (presión, humedad, temperatura, iradiancia solar, evotranpiración, pluviometría, dirección del viento, etc.).
- Calidad del aire.
- Nivel de llenado de los contenedores.

También se puede ver información de algunos activos energéticos como instalaciones soleres térmicas y vehículos de gestión de residuos adquiriendo todos los datos que pueden ofrecer estos activos.

Estas dos plataformas ofrecen:

- Ubicación por medio de un mapa de la localización de cada uno de los sensores.
- Visualización por medio de gráficos de los datos adquiridos por los sensores.
- Documentación API, para poder adquirir y hacer uso de esos datos si sabes cómo.

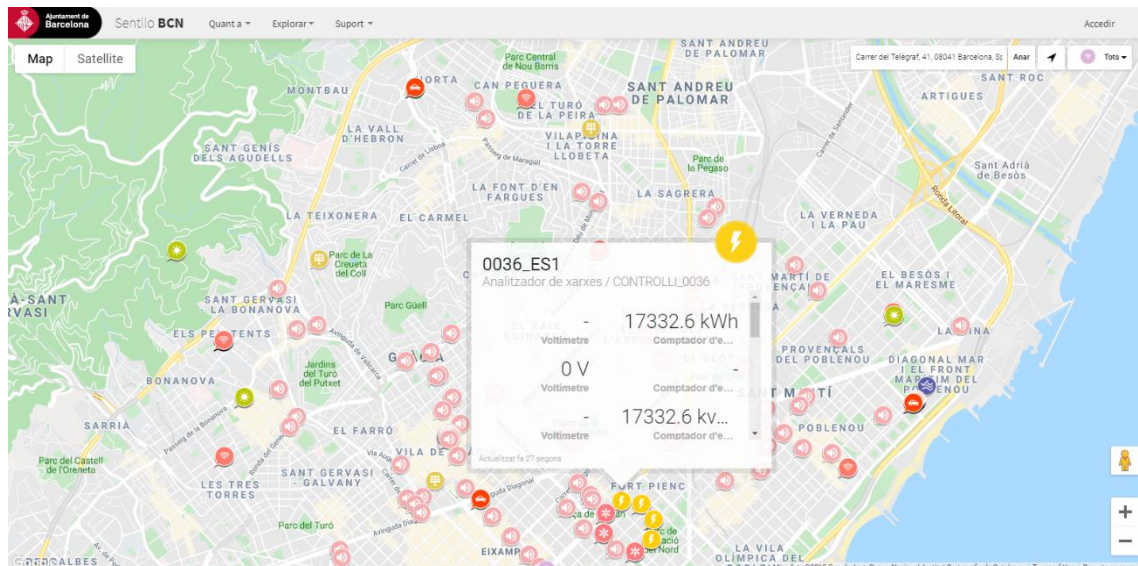


Figura 7.2. Sentilo ciudad de Barcelona. Fuente: Sentilo BCN

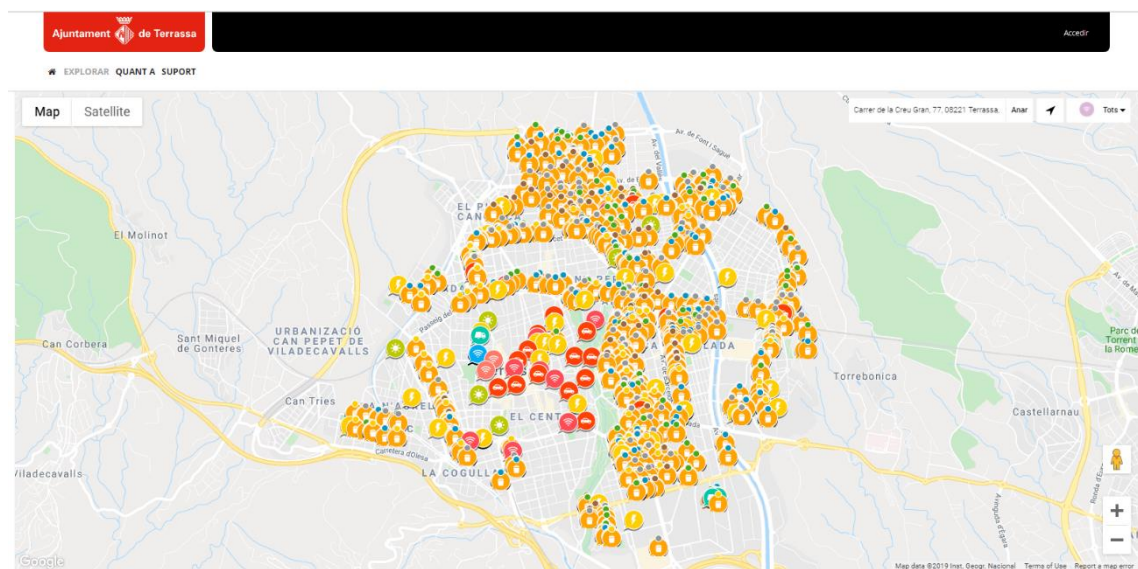


Figura 7.3. Sentilo ciudad de Terrassa. Fuente: Sentilo Terrassa

Fuentes:

- <http://sentilo.terrassa.cat/sentilo-catalog-web/component/map>
- <http://sentilo.bcn.cat/connecta-catalog-web/component/map#>

7.3. DEXMA. DEXCELL: Software para gestión energética.

Otro ejemplo de IoT aplicado en el sector energético es el que ha creado la empresa DEXMA, con su software de DEXCEL, de monitorización, análisis y control para el gestor energético.

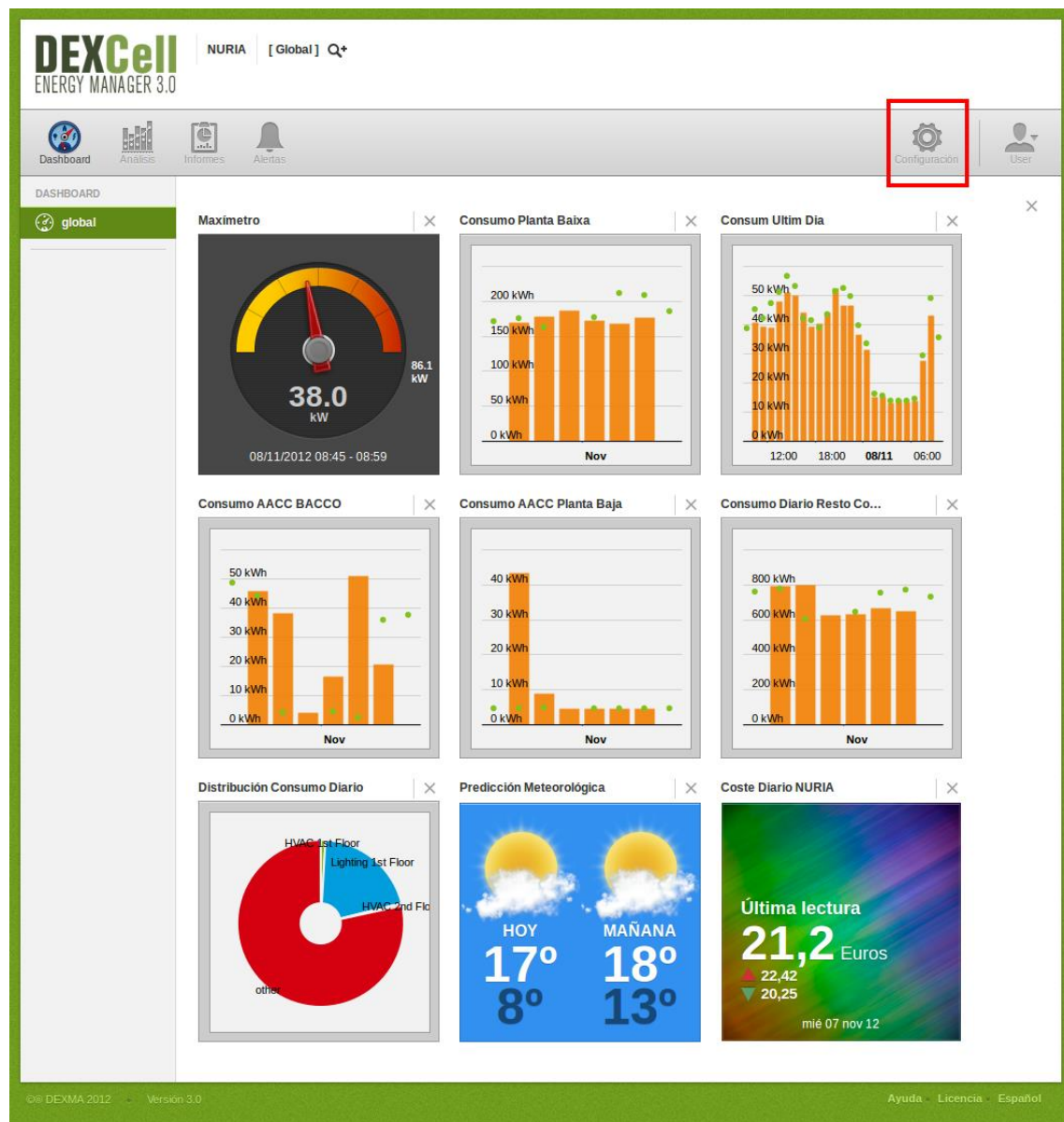


Figura 7.4. Panel de visualización de datos de Dexma. Fuente: DEXMA

Este software ofrece:

- Monitorización energética total.
- Análisis en tiempo real.
- Monitorización con cualquier tipo de hardware
- Y muchos más servicios.

Otras empresas que ofrecen servicios similares son, Circutor, Effilogics Technologies, Senion, etc.

7.4. Empresas y StartUps del sector IoT.

Para finalizar, a modo de pequeño estudio de mercado para ilustrar la utilidad que tiene el concepto de IoT y ver cómo se está aplicando en la vida real, se muestran algunas Empresa y StartUps que utilizan el concepto de IoT como modelo de negocio.

Tabla 13. Empresas aplicando el concepto de IoT. Fuente: Elaboración propia

Empresas aplicando el concepto de IoT	
https://thethings.io/	http://www.libelium.com/
https://miot.es/	http://www.creatio-control.com/index.php/es/
https://weathercloud.net/	https://www.integrainnovation.com/
https://www.useit.es/	https://www.amplia-iiot.com/
https://www.iotize.com/	https://www.sigfox.es/
https://iotied.com/	http://www.iotsens.com/es/
https://deepsolutions.io/	http://www.wyres.eu/
https://www.iotprojects.io/	https://zyfra.com/en/
https://thingstream.io/	https://www.firmware.org/
https://www.actility.com/	http://www.iotsens.com/es/
https://lesmartomation.com/	http://www.smart-cigo.com/smart-cigo/

8. Conclusiones de la parte teórica

Al haber leído los apartados anteriores, podemos hacernos una idea de que el IoT abarca muchas tecnologías, en el presente documento solo se han tocado lo relativo a la prueba de concepto a realizar en la fase experimental y también se han explicado de manera muy genérica otras tecnologías más.

Para realizar una solución IoT, ya sea una prueba de concepto o una solución completa para el mercado, es necesario conocer qué tecnología se ajustará más a nuestras necesidades. Es necesario hacerse algunas preguntas básicas apoyándonos en las capas que componen el concepto de IoT. Preguntas como:

- ¿Es relevante la velocidad de transmisión de los datos? ¿Qué es más conveniente, optar por una transmisión de datos inalámbrica o un por cable según el contexto?
- ¿Para la solución a ofrecer, es necesario una nube privada, pública o nos conviene que esté todo en local?
- ¿Con qué lenguajes vamos a construir la capa de presentación? ¿Qué información queremos mostrar? ¿Es necesario incluir en la solución técnicas de analítica de datos, es decir, Big Data e IA?
- ¿Cuál es el hardware que necesitamos para construir la solución? ¿Qué tipos de sensores serán necesarios?

El conocimiento es otro denominador común, a lo largo de todo el proyecto. Ha sido crucial saber que conocimientos es necesario adquirir para realizar la fase experimental.

También en los últimos apartados, se ha visto que existen empresas en la actualidad que ya están ofreciendo soluciones IoT e incluso algunas ciudades están aplicado el concepto de IoT junto con la Smart City. Según la curva de tendencias y expectativas (Hype Cycle), todavía quedan entre 5 a 10 años para que las soluciones de negocio IoT que se ofrecen en el mercado sean ampliamente implementadas y para que su lugar en el mercado y sus aplicaciones sean bien entendidas.

Ha todo esto se cree que es buen momento para continuar aprendiendo las tecnologías que involucran el concepto de IoT y continuar transformando los sectores ya definidos actualmente.

9. Biografía, referencias y recursos utilizados de la parte teórica

Bibliografía:

- [1] Aguilar, L. J. (2017). *Industria 4.0. La revolución industrial*. México: Alfaomega, Marcombo.
- [2] García, S. C. (2014). *Smart City. Hacia la gestión inteligente*. Barcelona: Marcombo.
- [3] Molloy, D. (2018). *Raspberry Pi a fondo para desarrolladores*. Barcelona: Marcombo.
- [4] Wikipedia. (22 de 11 de 2019). *Modelo OSI*. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Modelo_OSI
- [5] Wikipedia. (03 de 12 de 2019). *Modelo TCP/IP*. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Modelo_TCP/IP
- [6] mntolia. (s.f.). *Fundamentals of MQTT*. Obtenido de mntolia:
<https://mntolia.com/fundamentals-mqtt/>
- [7] Schmarzo, B. (2014). *Big Data. El poder de los datos*. Madrid: Grupo Anaya,S.A.
- [8] Seuba, M. L. (2019). *Internet de las Cosas. La transformación digital de la sociedad*. Barcelona: Ra-Ma.
- [9] Barcelona Activa. (18 de 06 de 2018). Integració de tecnologies "Internet of Things" en la pime industrial. Barcelona.
- [10] Ravulavaru, A. (2017). *Practical Internet of Things with JavaScript*. Birmingham: Packt>.

Recursos

<https://www.lucidchart.com>
<https://aws.amazon.com/es/>
<https://drive.google.com/>

BLOQUE EXPERIMENTAL: CREACIÓN DE LA PRUEBA DE CONCEPTO (PdC)

1. Construcción de la plataforma IoT.

Este bloque corresponde a la segunda parte de este proyecto, en el que se deja de explicar la teoría y el concepto de IoT y se pasa a explicar paso a paso cómo implementar una prueba de concepto IoT.

En los apartados siguientes, se explica cómo crear una arquitectura básica del concepto IoT utilizando todo lo explicado anteriormente en el bloque teórico.

La arquitectura básica a explicar es una simple página web donde se puede ver los datos de captura en tiempo real de un sensor de DHT22, que captura la humedad y la temperatura del entorno donde se encuentra.

Para poder ver el resultado final de esta arquitectura básica se facilitan los siguientes recursos:

- El resultado final de esta arquitectura básica se puede ver en el siguiente link (hacer clic en la opción de arquitectura básica):

<http://iotcloudsolutions.com/>

- Para obtener y poder ver el código completo se puede ir al siguiente link:

https://github.com/Richardbmk/iot_tfg

Para finalizar y ofrecer una prueba de concepto más completa se decide escalar la arquitectura básica poniendo más datos de sensores (humedad, temperatura, calidad del aire, presión, medidor de energía). Esta arquitectura, no se explica y simplemente se ofrece las ideas generales y todos los recursos para crear, por uno mismo, la prueba de concepto si se desea.

Para poder ver el resultado final de esta arquitectura básica se facilitan los siguientes recursos:

- El resultado final de esta arquitectura se puede ver en el siguiente link (hacer clic en la opción de arquitectura definitiva):

<http://iotcloudsolutions.com/>

- Para obtener y poder ver el código completo se puede ir al siguiente link:

https://github.com/Richardbmk/iot_tfg

Las herramientas utilizadas para la creación de esta prueba de concepto son las siguientes:

- Herramientas Hardware utilizadas:
 - Raspberry Pi 3 Model B+ y DHT22 Sensor
- Herramientas software utilizadas:
 - BackEnd:
 - AWS IoT, AWS SDK para Python, AWS S3, AWS IAM, Python (Dentro de la Raspberry Pi) y AWS DynamoDB (Base de datos)
 - FrontEnd:
 - CSS (Bootstrap), HTML, JavaScript and ChartJS

La elección de estos lenguajes de programación y el proveedor de servicios en la nube se ha hecho básicamente por disponibilidad de recursos online (blog, tutoriales, documentación oficial), reducción de costes, facilidad de aprendizaje y verificación de hipótesis acudiendo a la feria anual de [IoT Solutions World Congress](#).

2. Preparación de la RPi como dispositivo IoT

En este apartado se realizarán las preparaciones necesarias para que la Raspberry Pi (RPi) junto los sensores, pueda enviar los datos capturados a la nube, preparándolo de esta manera para actuar como un dispositivo IoT. Para esta preparación previa será necesario lo siguiente:

1. Instalar el sistema operativo a la RPi, ya que partimos de un dispositivo nuevo.
2. Conectividad a Internet vía Wi-Fi
3. Conectividad a Internet vía cable Ethernet (Opcional)
4. Comunicación con la RPi con el ordenador personal vía SSH (Secure Shel) mediante PuTTY
5. Configuración básica de la RPi, preparación del entorno virtual e instalaciones de todo el software necesario.

Como ya se ha mencionado anteriormente, utilizaremos la RPi en modo Headless por lo que el material necesario será el justo y necesario para realizar la función de envío de datos. No dispondremos de pantalla, ratón ni teclado para la RPi. El material utilizado para este primer paso es el siguiente:

- Raspberry Pi Modelo 3 B+
- Fuente de alimentación para la RPi
- Tarjeta Micro-SD de 32 GB
- Cable Ethernet CAT-5 o superior (CAT-6, CAT-6A) (opcional)
- Lector de tarjetas micro-SD
- Ordenador personal con Windows 7 como sistema operativo u otro que se sepa utilizar.

2.1. Instalación el sistema operativo a la RPi.

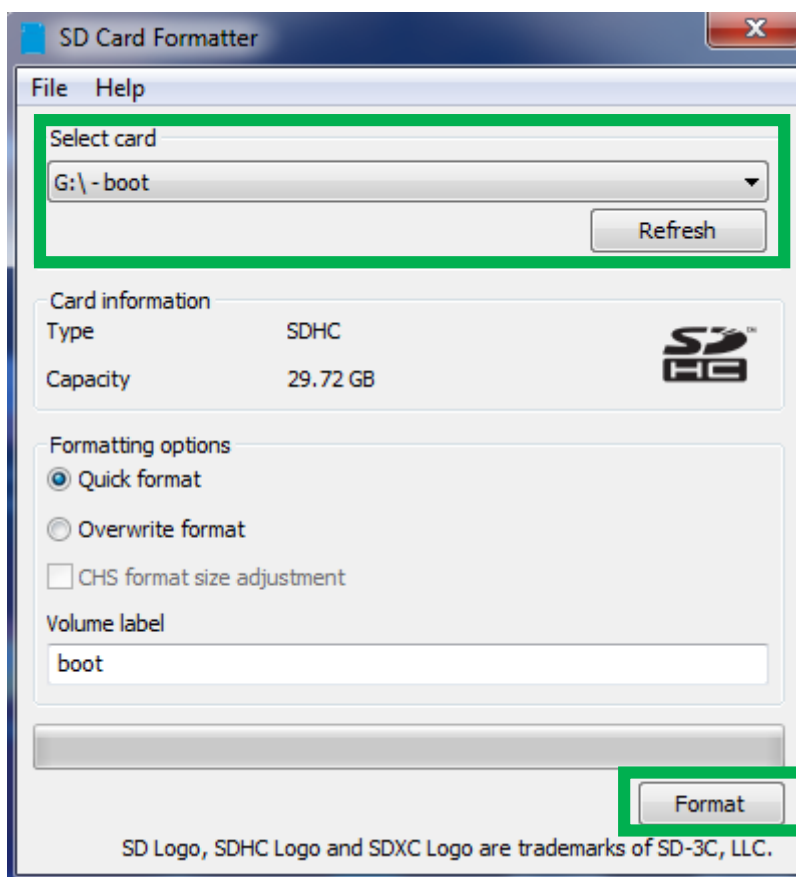
Con la ayuda de un lector de tarjetas conectamos la tarjeta micro-SD de 32 GB a nuestro ordenador, dependiendo de dónde hayamos comprado la tarjeta micro-SD o si ya venía incluida con la RPi, es posible que ya haya algún archivo dentro.

Normalmente si compramos la RPi con la tarjeta micro-SD viene con un archivo NOOBS para Raspberry Pi, que no es más que un instalador de Linux que permite de forma fácil instalar los sistemas operativos que vienen por defecto en el propio NOOBS (Raspbian y LibreELEC). Este método no es el que utilizaremos, ya que necesitaremos un sistema operativo ligero para cumplir con el concepto de headless systems.

Por lo tanto, formatearemos la tarjeta micro-SD venga con lo que venga de fábrica. Para ello utilizaremos el software SD Card Formatter:

Pasos para formatear la tarjeta micro-SD:

1. Descargar el software "SD Card Formatter".
2. Conectar la tarjeta al ordenador personal
3. Abrir el programa informático
4. Seleccionar la tarjeta a formatear en el programa y dejar las demás opciones por defecto.
5. Pulsar formatear



En la figura 2.1, podemos hacernos una idea de simple proceso de formatear la tarjeta micro-SD.

Una vez realizado este pequeño paso estamos preparados para instalar el sistema operativo necesario para el proyecto.

Es el mismo proceso que se sigue para instalar cualquier otro sistema operativo que sea necesario por la razón que sea.

Figura 2.1. Formateo de la tarjeta micro-SD. Fuente: Pantallazo PC

Pasos para instalar el Sistema Operativo (SO):

1. A través de la página oficial nos descargamos la versión del sistema operativo necesario (Raspbian Buster Lite). Página web: <https://www.raspberrypi.org/downloads/raspbian/>
2. Descargamos e instalamos en el ordenador personal el software “Win32 Disk Imager”, necesario para crear una imagen del SO elegido en la tarjeta micro-SD.
3. Una vez realizado los dos pasos anteriores, abrimos el software Win32 Disk Imager y realizamos los siguientes pasos:
 - a. Con la venta del software, seleccionamos la opción “Divice” y seleccionamos la tarjeta micro-SD.
 - b. En la opción “Image File”, elegimos dónde tenemos guardado el archivo y a la vez le damos un nombre significativo como, por ejemplo: *RPi-Buster-Lite-withWireLessConfigHOME-01082019.img*
 - c. Dejamos todas las demás opciones por defecto y por último le damos al botón Write.
4. Para el caso que quisiéramos realizar una copia de seguridad, práctica de uso muy recomendada, los pasos del punto 3 serías similares solo haríamos diferente el paso c), en vez de el botón Write pulsáramos el botón Read, que significa leer la información de la tarjeta micro-SD.

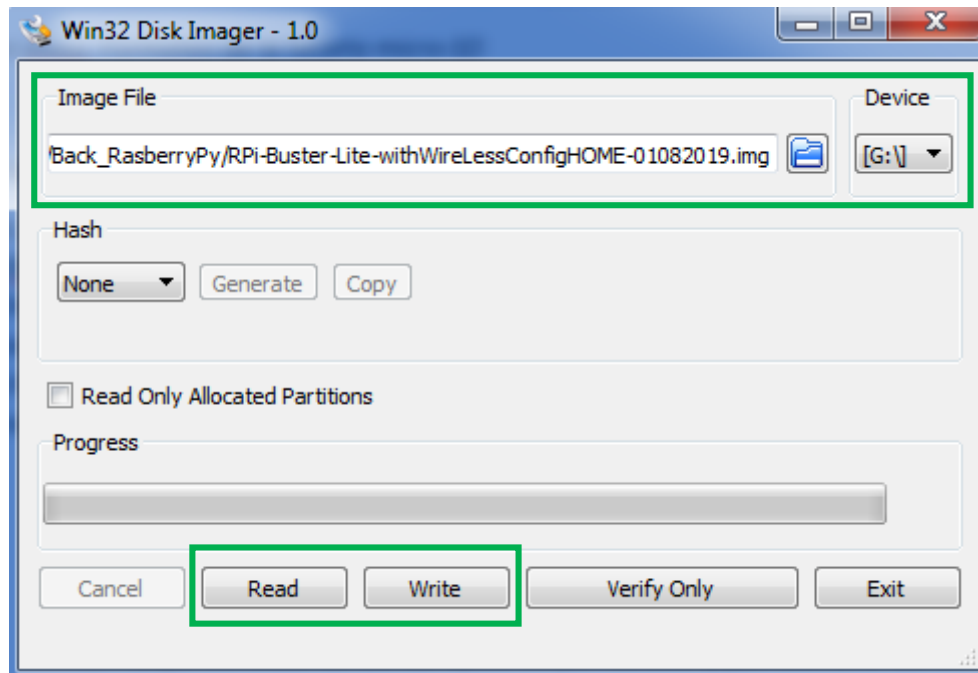


Figura 2.2. Creación de la imagen en la tarjeta micro-SD. Fuente: Pantallazo PC

En la Figura 2.2, podemos hacernos una idea del proceso.

En estos momentos ya tenemos instalado el sistema operativo en nuestra tarjeta micro-SD y ya estamos preparados para conectarlo a la Raspberry Pi y proceder a realizar los siguientes pasos.

Como ya hemos explicado en la parte teórica del presente documento, es posible conectar la RPi a internet vía Wi-Fi o con un cable Ethernet.

En el presente proyecto utilizaremos la conexión Wi-Fi con preferencia, la conexión por cable Ethernet se utilizará en los casos que no sea posible conectarse a internet por Wi-Fi, por lo que puede ser conveniente disponer de un cable de este tipo.

En los apartados siguiente, se procede a explicar cómo realizar estas dos configuraciones para poder conectarnos a internet.

2.2. Conectividad a Internet vía Wi-Fi

Para realizar esta fase de manera ágil y rápida, sin desconectar todavía del ordenador personal la tarjeta micro-SD una vez instalado el sistema operativo necesario, seguiremos los siguientes pasos:

1. Entrar en los archivos de la tarjeta micro-SD y crear los siguientes archivos:
 - a. Crear un archivo con el nombre `ssh`, sin extensión alguna para habilitar SSH (Secure Shell). Esto será necesario para poder comunicarnos más adelante con nuestra RPi a través de nuestro ordenador personal.
 - b. Para habilitar el Wi-Fi, crear un archivo con el nombre `"wpa_supplicant.conf"` con el siguiente contenido:

```
country=ES

ctrl_interface=DIR=/var/run/wpa_supplicant
GROUP=netdev

update_config=1

network={

    ssid="MOVISTAR_XXXX"

    psk="XXXXXXXXXXXXXXXXXXXX"

    key_mgmt=WPA-PSK

}
```

Script 2.1. Configuración Wi-Fi RPi.

Será necesario remplazar *country*, *ssid* y *psk* con los datos que correspondan según nuestro país, el nombre de nuestro Router y la contraseña.

Una vez realizados estos dos pasos, ya estamos preparados para utilizar la RPi, con conexión Wi-Fi. Basta con desconectar la tarjeta micro-SD de nuestro ordenador personal y ponerlo en la RPi para comenzar a utilizarlo.

Más adelante se explica la comunicación con la RPi con conexión remota.

2.3. Conectividad a Internet vía cable Ethernet

Este modo de conexión es muy sencillo comparada con la anterior y basta con desconectar la tarjeta micro-SD de nuestro ordenador y colocarlo en la RPi. Una vez realizado eso, podemos proceder a conectar el cable Ethernet en el Router de nuestro ordenador personal y el otro extremo del cable a la entrada RJ45 de la RPi, en otras palabras, conectar a internet con un cable Ethernet como lo haríamos con nuestro ordenador de sobremesa.

Y con esto ya tendríamos de internet y podemos acceder a nuestra RPi vía conexión remota, que procederemos a explicar a continuación.

2.4. Comunicación con la RPi con el ordenador personal vía SSH (Secure Shel) mediante PuTTY

Para poder comunicarnos con la RPi a través de nuestro ordenador personal, con el sistema operativo instalado será necesario los dos siguientes programas: Advanced IP Scanner y PuTTY.

Advanced IP Scanner, nos será de utilidad para detectar la dirección IP que ha sido asignada a nuestra RPi por nuestro Router. Tendremos una dirección IP para la conexión vía Wi-Fi y una muy similar para la conexión vía Ethernet. Por lo que procederemos a descargar e instalar este programa informático.

En la figura 2.3, podemos ver el software. Basta con darle al botón explorar para que nos muestre todas las direcciones IP de todos los dispositivos conectados a nuestro ordenador. Si

nos fijamos en esta misma imagen podemos ver que tenemos dos direcciones IP asignadas a la RPi. La dirección IP 192.168.1.90 corresponde a la conexión vía Wi-Fi y la 192.168.1.89 corresponde a la conexión vía Ethernet.

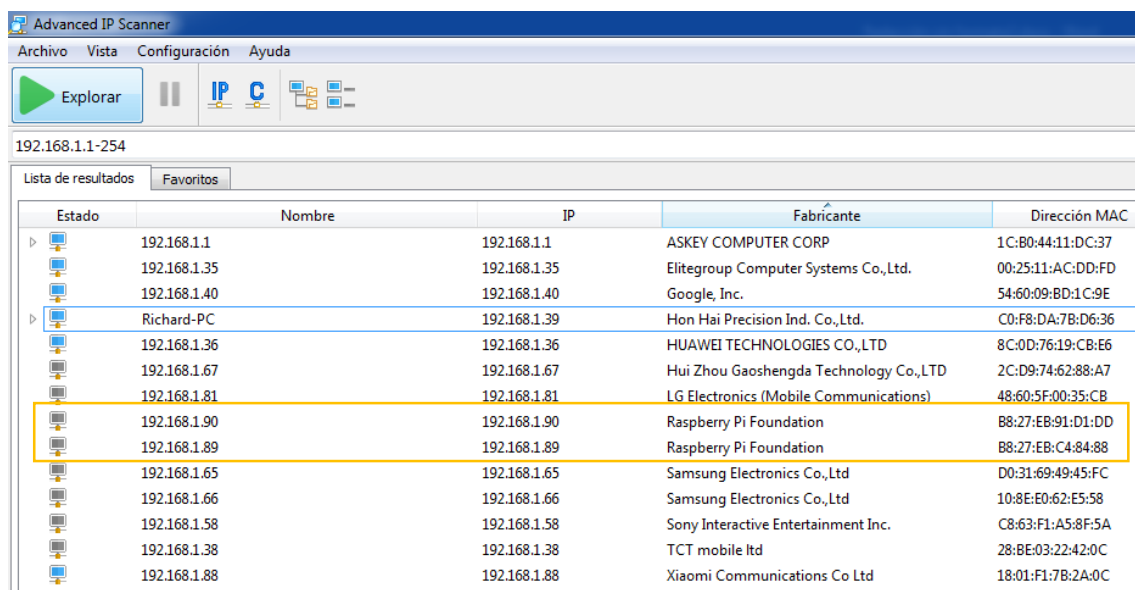


Figura 2.3. Advanced IP Scanner. Fuente: Pantallazo PC

Una vez conocida la dirección IP de los dos tipos de conexión podemos elegir con qué modo de conexión queremos comunicarnos con la RPi. Seguiremos con el modo de conexión vía Wi-Fi, por lo que la dirección IP que utilizaremos será la 192.168.1.90.

Ahora procedemos a descargar e instalar el software PuTTY. PuTTY es un programa informático que nos permitirá comunicarnos vía SSH con la RPi una simple ventana de comandos, como la que tenemos en nuestro sistema operativo Windows 7.

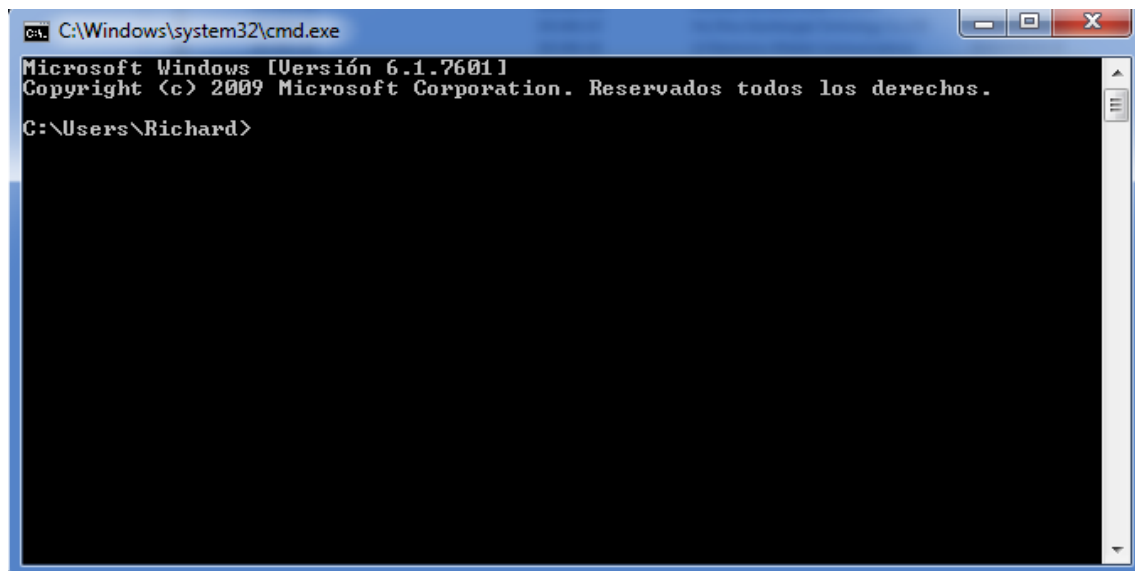


Figura 2.4. Ventana de comando Windows 7. Fuente: Pantallazo PC

Ahora que ya tenemos el programa PuTTY instalado en nuestro ordenador, procedemos a configurar el programa para abrir una terminal que nos permita comunicarnos con la RPi.

Pasos a seguir:

1. Abrir el programa PuTTY
2. Copiar y pegar la dirección IP de modo de conexión deseado.
3. Pulsar Open

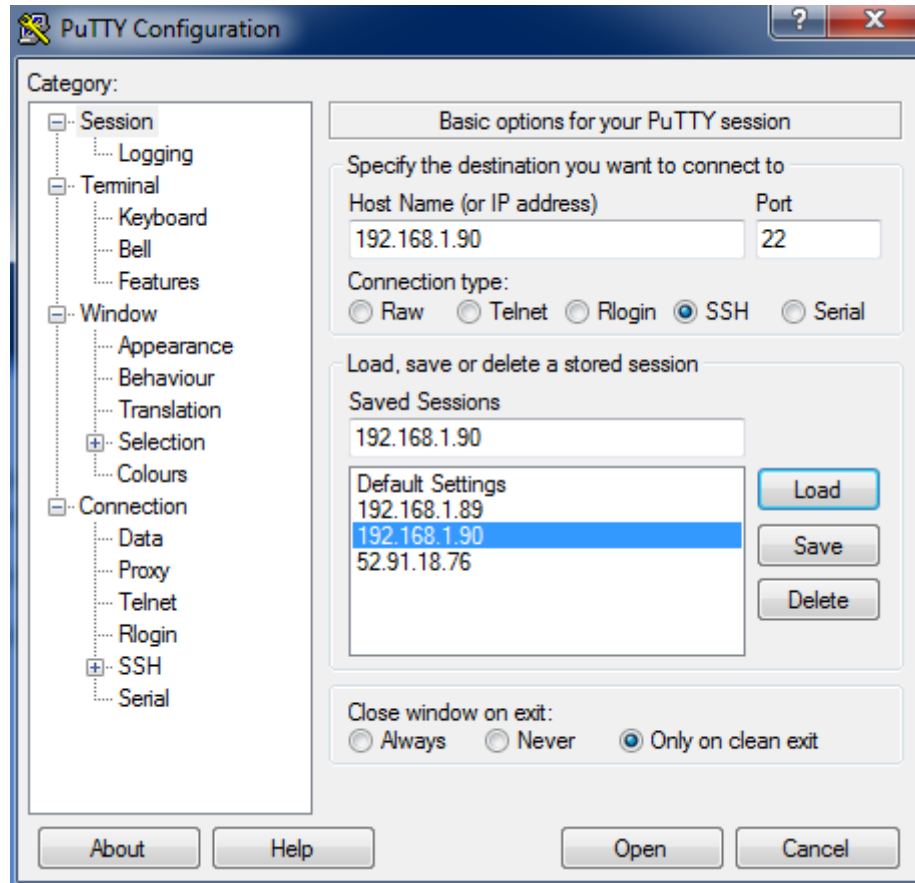


Figura 2.5. PuTTY, abriendo el terminal. Fuente: Pantallazo PC

Así de simple. Esto nos abrirá la terminal que se muestra en la Figura 2.6.

Con esto ya estaríamos preparados para comenzar a comunicarnos con la RPI a través de nuestro ordenador personal. En la Figura 2.6, podemos ver que se nos solicita el nombre de usuario (login as) y la contraseña.

Al instalar un sistema operativo por primera vez en la tarjeta micro-SD, se nos asigna por defecto los siguientes datos de entrada:

Login as: pi

pi@192.168.1.90's password: raspberry

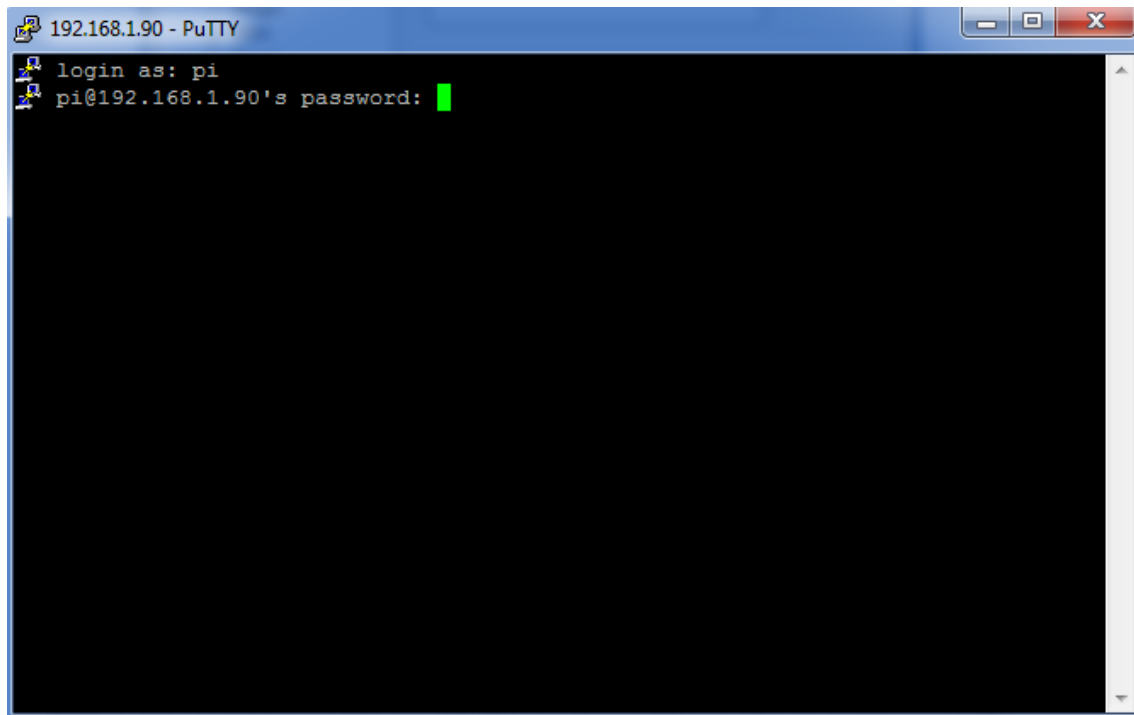


Figura 2.6. Ventana de comando con PuTTY. Fuente: Pantallazo PC

Una buena práctica de seguridad es cambiar la contraseña por defecto, por lo que para cumplir con los principios de seguridad procederemos a entrar y cambiar la contraseña de acceso a la RPi.

Pasos para cambiar la contraseña por defecto de la RPi:

1. En la ventana de comando de PuTTY, una vez dentro del sistema operativo de la RPi habiendo puesto el usuario y contraseña correctamente, es necesario escribir el siguiente comando: `sudo raspi-config`. Pulsamos INTRO y se nos abrirá la ventana de la Figura 2.7.
2. La primera opción (*Change User Password Change password for the current user*) ya nos permite cambiar la contraseña, basta con pulsar nuevamente INTRO y se nos pedirá que introduzcamos una nueva contraseña.
3. Basta con seguir los pasos y ya tendremos una nueva contraseña para la RPi.

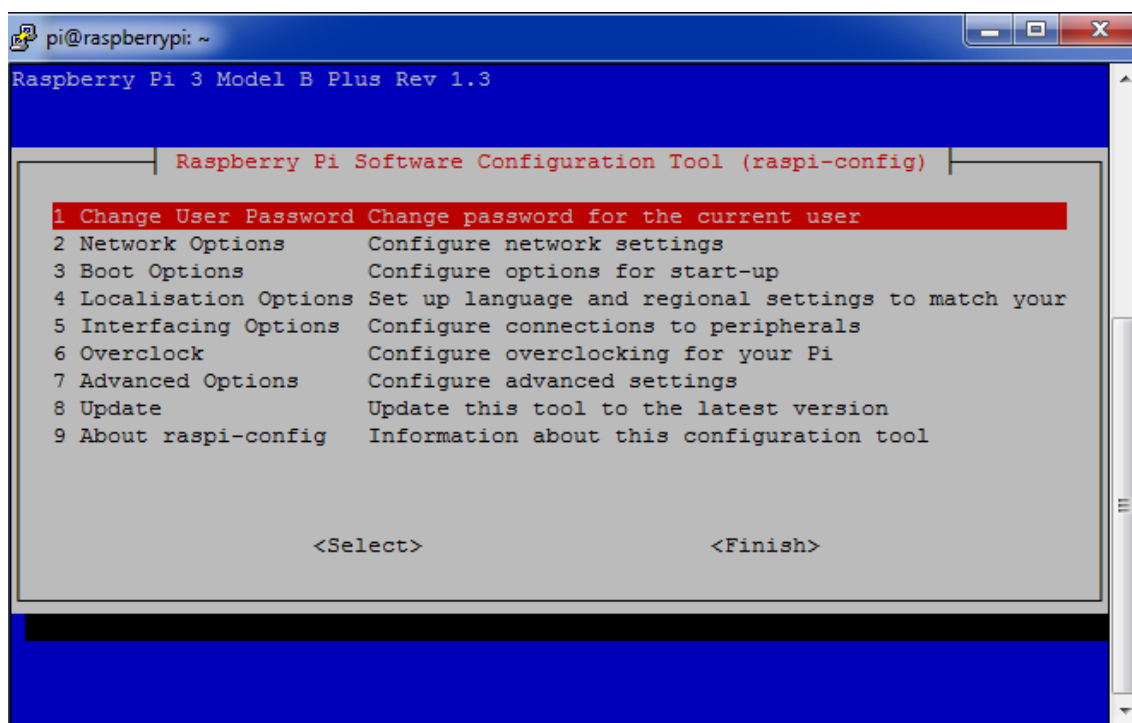


Figura 2.7. Ventana de configuración de la RPi. Fuente: Pantallazo PC

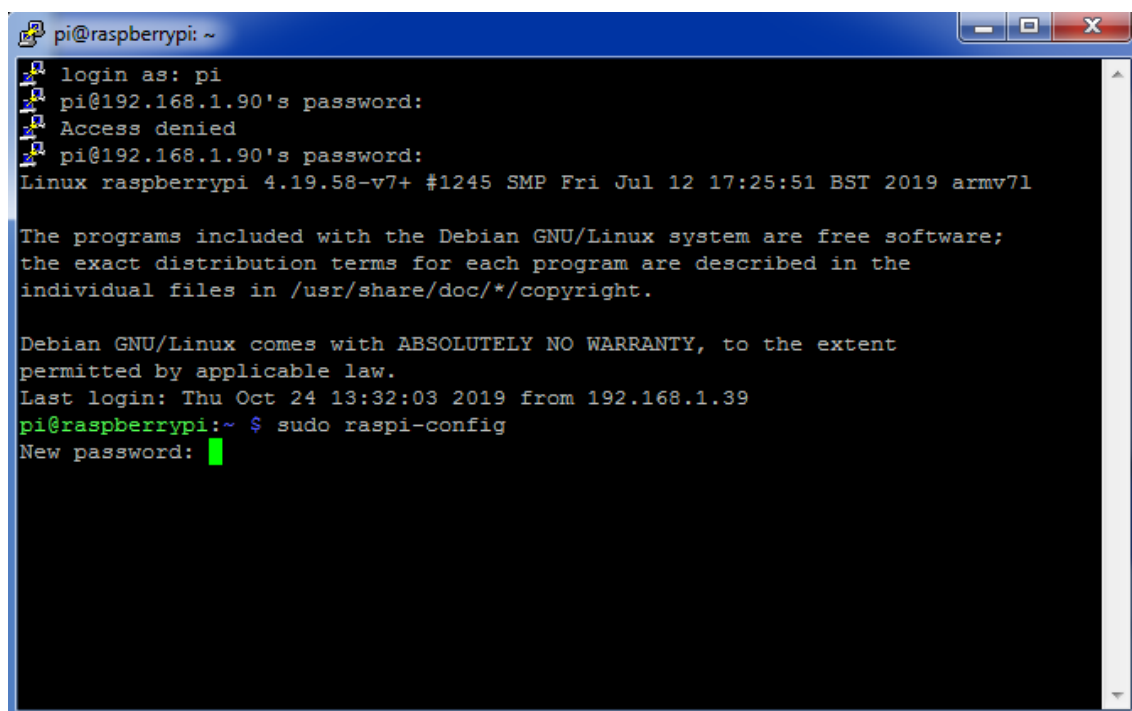


Figura 2.8. Cambio de contraseña de la RPi. Fuente: Pantallazo PC

2.5. Configuración básica de la RPi, preparación del entorno virtual e instalaciones de todo el software necesario.

Una vez ya tenemos la contraseña cambiada, ya estamos en disposición de poder instalar todo el software necesario para convertir nuestra RPi como un dispositivo IoT. Las instalaciones que se explican a continuación son:

- La instalación de un entorno virtual
- Instalación de las actualizaciones
- Instalación de los módulos y paquetes necesarios
- Instalación de FileZilla en el ordenador personal

La instalación de un entorno virtual.

Antes de proceder a la instalación conviene saber primero, ¿qué es un entorno virtual en Python? Y ¿Por qué es necesario para nuestra aplicación?

Un entorno virtual, es una herramienta que ayuda a mantener separadas las dependencias (módulos y paquetes) requeridas por diferentes proyectos al crear entornos virtuales de Python aislados para ellos. Cada aplicación, realizada con la RPi, tendrá su propio entorno virtual en el cual se le instala lo necesario y no interfiere con otras aplicaciones que creemos en el caso que sea necesario.

La razón por la que se crea un entorno virtual, es porque se está realizando una prueba de concepto (PdC) y la RPi no servirá única y exclusivamente como dispositivo IoT, por lo que es conveniente aislar los módulos y paquetes de otras aplicaciones que creemos en el futuro para la RPi. En la siguiente Figura 2.9, se puede ver ilustrado este concepto.

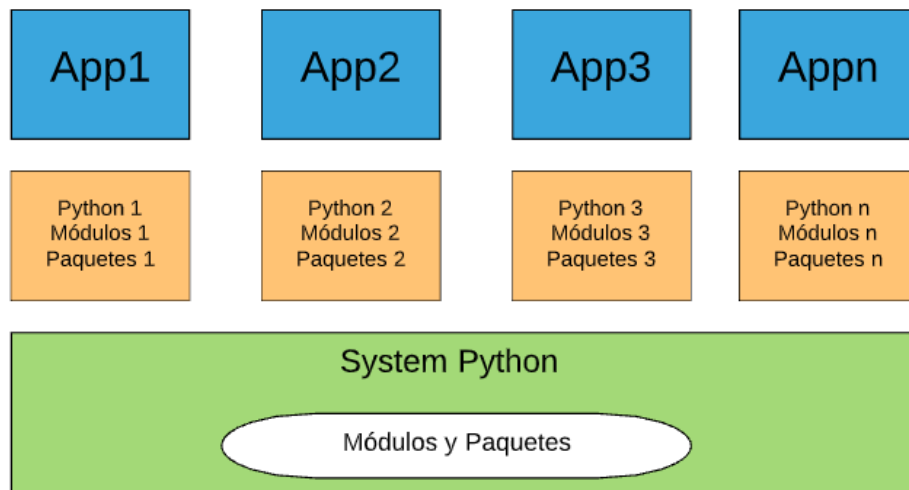


Figura 2.9. Instalación de paquetes en el entorno virtual correspondiente. Fuente: elaboración propia

A continuación, se explica paso a paso cómo crear este entorno virtual que nos permitirá instalar todos los módulos, paquetes e incluso un nuevo lenguaje de servidor diferente de Python como puede ser nodejs si es necesario. Partiendo de la una línea de comando como la Figura 2.10, y sabiendo que todo lo que hay actualmente en el sistema operativo son softwares instalados por defecto, procedemos a realizar los siguientes pasos:

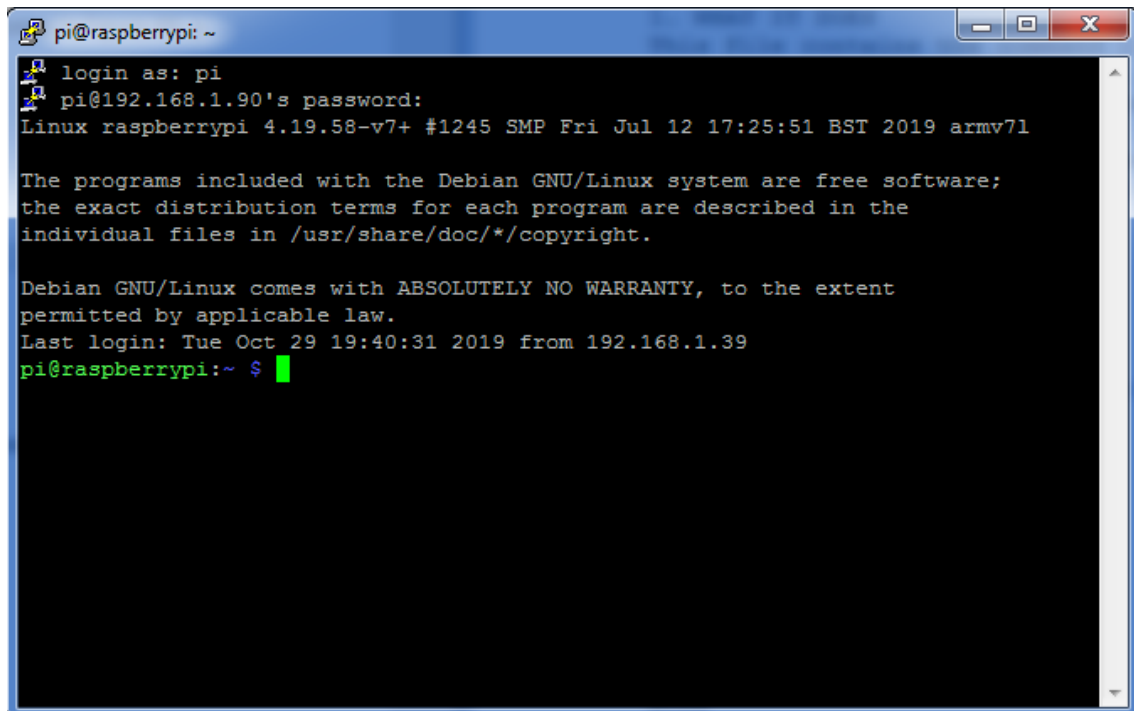


Figura 2.10. Línea de comandos RPi a través de PuTTY. Fuente: Pantallazo PC

Paso 1. Configuración del sistema.

- a) Escribimos la siguiente instrucción para actualizar todos los paquetes disponibles en el sistema operativo: `pi@raspberrypi: ~ $ sudo apt-get update`
- b) Escribimos la siguiente instrucción para instalar la versión más reciente de los paquetes disponibles en el sistema operativo: `pi@raspberrypi: ~ $ sudo apt-get upgrade`
- c) Instalamos el siguiente paquete escribiendo la siguiente instrucción: `pi@raspberrypi: ~ $ sudo apt-get install build-essential`

Lo que hace este paquete, es instalar una lista informativa de los paquetes que son esenciales para la creación de nuevos paquetes.

- d) Instalamos un conjunto de paquetes escribiendo la siguiente instrucción: `pi@raspberrypi: ~ $ sudo apt-get install libncurses5-dev libncursesw5-dev libreadline6-dev libffi-dev`
- e) Instalamos un conjunto de paquetes escribiendo la siguiente instrucción: `pi@raspberrypi: ~ $ sudo apt-get install libbz2-dev libexpat1-dev liblzma-dev zlib1g-dev libsqlite3-dev libgdbm-dev tk8.5-dev libssl-dev openssl`
- f) Instalamos las herramientas de desarrollo de Python con la siguiente instrucción: `pi@raspberrypi: ~ $ sudo apt-get install python-dev`

Paso 2. Instalación de Python 3

En la raíz del directorio es necesario crear una carpeta donde almacenar la nueva versión disponible de Python 3 e instalar dentro la nueva versión de Python3 (3.7.4). Realizaremos esto, siguiendo las siguientes instrucciones:

- a) Comprobar la versión actual de Python:
`pi@raspberrypi: ~ $ python3 -version`
- b) Para dirigirse a la raíz: `pi@raspberrypi: ~ $ cd ~`
- c) Crear una carpeta o directorio:
`pi@raspberrypi: ~ $ mkdir python-source`
- d) Entrar en la carpeta que se ha creado:
`pi@raspberrypi: ~ $ cd python-source/`
- e) Descargar la fuente del archivo con Python 3.7.4:
`pi@raspberrypi:~$
wget https://www.python.org/ftp/python/3.7.4/Python-3.7.4.tgz`
- f) Extraer el contenido de archivo descargado anteriormente:
`pi@raspberrypi:~$ tar zxvf Python-3.7.4.tgz`
- g) Acceder al directorio (carpeta) del nuevo Python 3.7.4:
`pi@raspberrypi:~$ cd Python-3.7.4/`
- h) Dentro de la carpeta Python-3.7.4, hay un archivo con el nombre “configure”, es necesario instalar lo que se encuentra dentro de ese archivo con la siguiente instrucción:
`pi@raspberrypi:~$
./configure --prefix=/usr/local/opt/python-3.7.4`
- i) Compilar con la siguiente instrucción:
`pi@raspberrypi:~$ make`
- j) Instalar la nueva versión de Python (Python 3.7.4):
`pi@raspberrypi:~$ sudo make install`
- k) Comprobar si se ha instalado la nueva versión correctamente. La repuesta del sistema debe ser Python 3.7.4:
`pi@raspberrypi:~$
/usr/local/opt/python-3.7.4/bin/python3.7 --version`

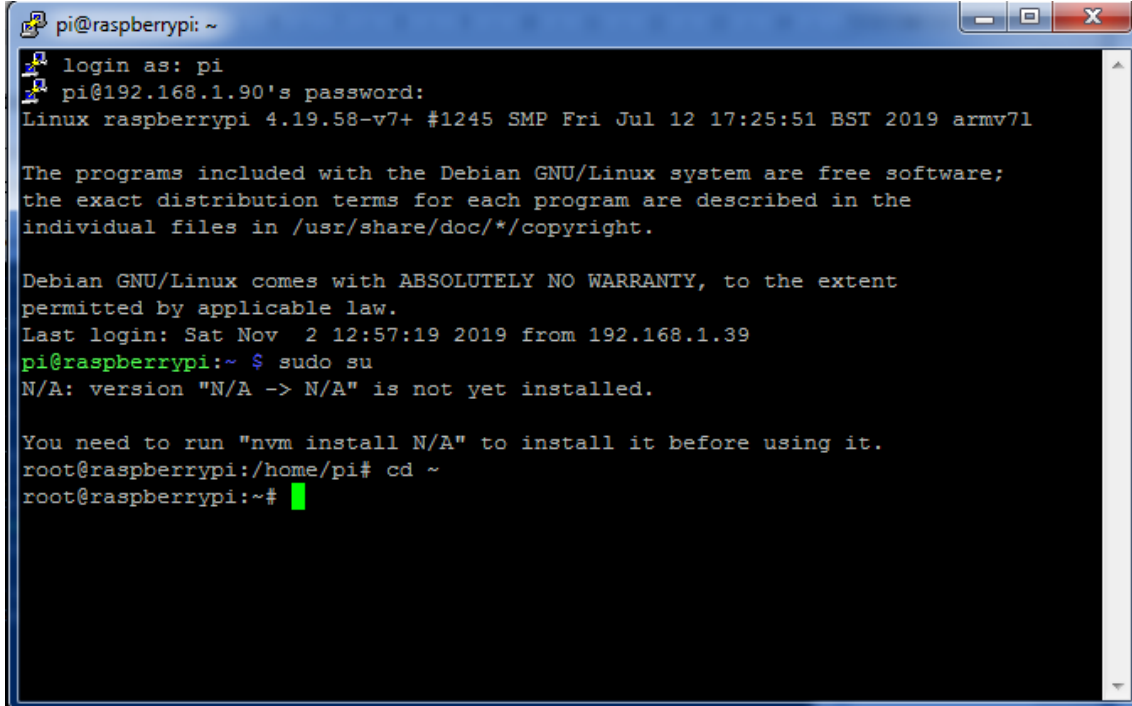
Recordar que la nueva versión que se ha instalado, se ha hecho dentro del propio sistema operativo y no en un entorno virtual. También aclarar que no se ha sobre-escrito la versión anterior de Python, por lo que coexisten la versión que viene por defecto en el sistema operativo y la que se ha instalado.

Llegados a este punto es recomendable hacer una copia de seguridad.

Paso 3. Instalación del entorno virtual

- a) Para realizar las siguientes instrucciones es necesario hacerlo como raíz por lo que debemos ejecutar la siguiente instrucción: `pi@raspberrypi: ~ $ sudo su`

La terminal debe estar igual que se muestra en la Figura 2.11



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.1.90's password:  
Linux raspberrypi 4.19.58-v7+ #1245 SMP Fri Jul 12 17:25:51 BST 2019 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sat Nov  2 12:57:19 2019 from 192.168.1.39  
pi@raspberrypi:~ $ sudo su  
N/A: version "N/A -> N/A" is not yet installed.  
  
You need to run "nvm install N/A" to install it before using it.  
root@raspberrypi:/home/pi# cd ~  
root@raspberrypi:~#
```

Figura 2.11. Acceso como raíz. Fuente: Pantallazo PC

- b) Creamos los directorios necesarios en el lugar donde deseamos instalar el entorno virtual para nuestra aplicación:

- a. `root@pi@raspberrypi:~# mkdir /var/awsiot`
- b. `root@pi@raspberrypi:~# mkdir /var/www/tfg_iot/`
- c. `root@pi@raspberrypi:~# cd /var/www/lab_app/`

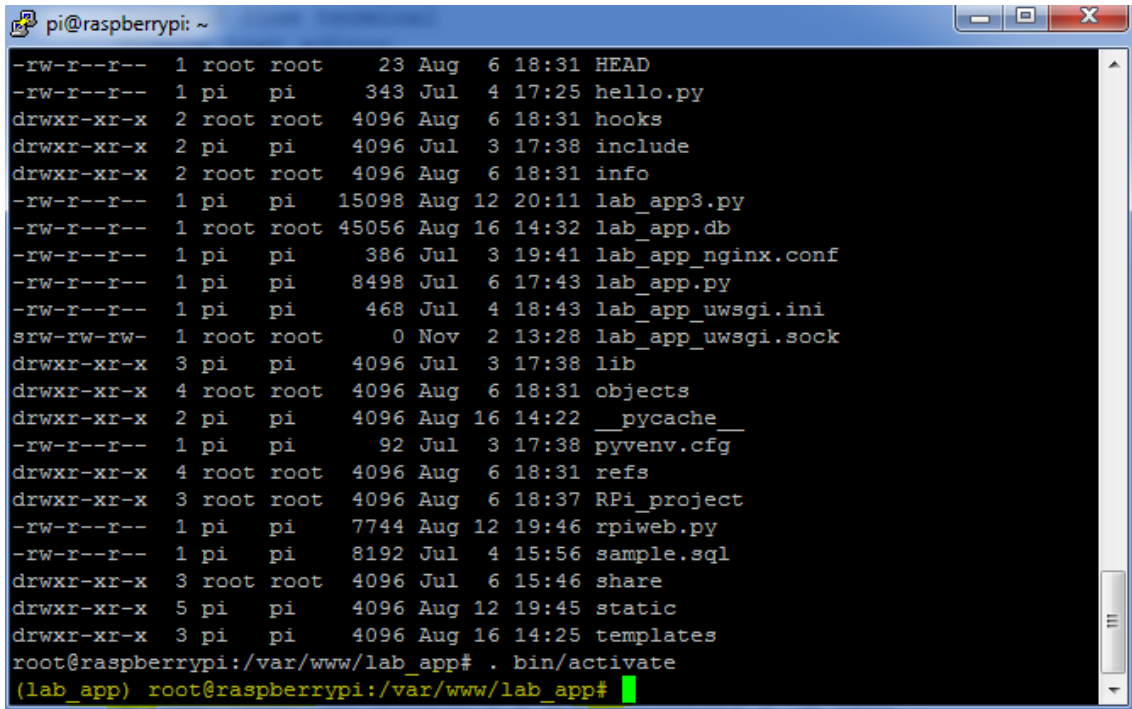
- c) Creación del entorno virtual e instalación:

```
root@pi@raspberrypi:~# /usr/local/opt/python-  
3.6.4/bin/python3.6 -m venv .
```

(El punto sirve para indicar que la instalación se hace en el mismo directorio donde te encuentras)

- d) Activar el entorno virtual:

```
root@pi@raspberrypi:/var/www/lab_app# . bin/activate
```



```

pi@raspberrypi: ~
-rw-r--r-- 1 root root 23 Aug 6 18:31 HEAD
-rw-r--r-- 1 pi pi 343 Jul 4 17:25 hello.py
drwxr-xr-x 2 root root 4096 Aug 6 18:31 hooks
drwxr-xr-x 2 pi pi 4096 Jul 3 17:38 include
drwxr-xr-x 2 root root 4096 Aug 6 18:31 info
-rw-r--r-- 1 pi pi 15098 Aug 12 20:11 lab_app3.py
-rw-r--r-- 1 root root 45056 Aug 16 14:32 lab_app.db
-rw-r--r-- 1 pi pi 386 Jul 3 19:41 lab_app_nginx.conf
-rw-r--r-- 1 pi pi 8498 Jul 6 17:43 lab_app.py
-rw-r--r-- 1 pi pi 468 Jul 4 18:43 lab_app_uwsgi.ini
srw-rw-rw- 1 root root 0 Nov 2 13:28 lab_app_uwsgi.sock
drwxr-xr-x 3 pi pi 4096 Jul 3 17:38 lib
drwxr-xr-x 4 root root 4096 Aug 6 18:31 objects
drwxr-xr-x 2 pi pi 4096 Aug 16 14:22 __pycache__
-rw-r--r-- 1 pi pi 92 Jul 3 17:38 pyvenv.cfg
drwxr-xr-x 4 root root 4096 Aug 6 18:31 refs
drwxr-xr-x 3 root root 4096 Aug 6 18:37 RPi_project
-rw-r--r-- 1 pi pi 7744 Aug 12 19:46 rpiweb.py
-rw-r--r-- 1 pi pi 8192 Jul 4 15:56 sample.sql
drwxr-xr-x 3 root root 4096 Jul 6 15:46 share
drwxr-xr-x 5 pi pi 4096 Aug 12 19:45 static
drwxr-xr-x 3 pi pi 4096 Aug 16 14:25 templates
root@raspberrypi:/var/www/lab_app# . bin/activate
(lab_app) root@raspberrypi:/var/www/lab_app#

```

Figura 2.12. Entorno virtual activado. Fuente: Pantallazo PC

Si nos fijamos veremos en vez de “root@pi@raspberrypi:/var/www/lab_app#” ahora tenemos una inicial delante con el nombre del entorno virtual “(lab_app) root@raspberrypi:/var/www/lab_app#”.

Con el entorno virtual activado estamos en disposición de instalar todo lo que necesitamos dentro.

Si deseamos desactivar el entorno virtual basta con ejecutar la siguiente instrucción:

```
(lab_app) root@raspberrypi:/var/www/lab_app# deactivate
```

Paso 3. Instalación de Git.

Git es un sistema para almacenar códigos, librerías y otro tipo de datos. Instalarlo en la RPi nos permite clonar repositorios, es decir, hacer una copia en la RPi del archivo, código o librería que se ha guardado en el almacén de datos Git (Github). Para instalarla basta con poner las siguientes sentencias en la terminal:

```

pi@raspberrypi:~$ sudo apt-get install git-core
pi@raspberrypi:~$ git config --global user.email rick@gmail.com
pi@raspberrypi:~$ git config --global user.name "Rick"

```

Paso 4. Instalación de AWS IoT SDK para Python

AWS IoT SDK, es el paquete que nos permitirá realizar las comunicaciones via MQTT con Amazon Web Services. Partiendo del entorno virtual activado, para instalar este paquete es necesario seguir las siguientes instrucciones:

a) Clonar el repositorio en nuestra RPi:

```
(lab_app) root@raspberrypi:/var/www/lab_app#  
git clone https://github.com/aws/aws-iot-device-sdk-python.git
```

b) Acceder al archivo clonado e instalar AWS IoT SDK:

- a. `cd aws-iot-device-sdk-python`
- b. `python setup.py install`

Podríamos haber instalado simplemente este paquete siguiendo las instrucciones de la página oficial (<https://pypi.org/project/AWSIoTPythonSDK/1.0.0/>) y ya tendríamos lo necesario para utilizar la RPi como dispositivo IoT, pero como hemos explicado antes. No se utilizará la RPi solo para esta finalidad.

Llegados a este punto estamos más preparados para poder enviar los datos capturados por nuestro sensor a la nube de AWS y poder crear el aplicativo web. Para facilitar la transferencia de archivos entre la RPi y el ordenador personal, se recomienda instalar FileZilla.

Pasos para instalar FileZilla y comprender como realizar la transferencia de archivos entre la RPi y el ordenador personal:

1. Descargar el programa desde la página oficial:
<https://filezilla-project.org/download.php?type=client>
2. Instalar el programa

En la Figura 2.13, podemos hacernos una idea del funcionamiento. En la ventana marcada con el **número 1**, ponemos los datos para acceder a la RPi:

Servidor: `sftp:// 192.168.1.90`

Nombre de usuario: pi

Contraseña: *****

En la ventana **número 2**, tenemos el directorio del ordenador y en la ventana **número 3** tenemos el directorio de la RPi. Basta con arrastrar los archivos entre las diferentes ventanas para poder realizar la transferencia entre ordenador y RPi con comodidad.

Ya lo tenemos todo preparado. En el siguiente apartado avanzaremos un poco más para dar vida al concepto de IoT.

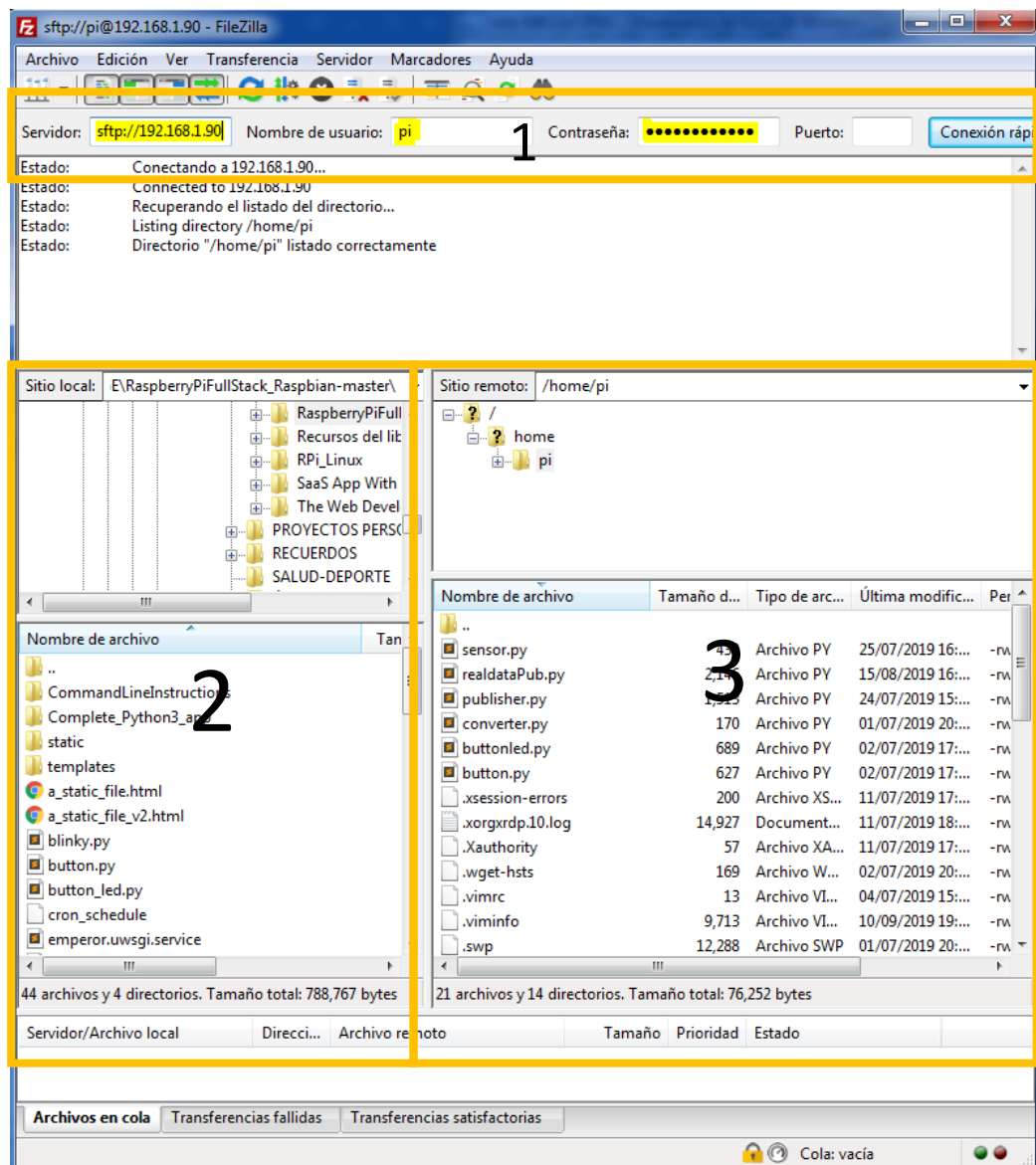


Figura 2.13. Ventana de FileZilla. Fuente: Pantallazo PC

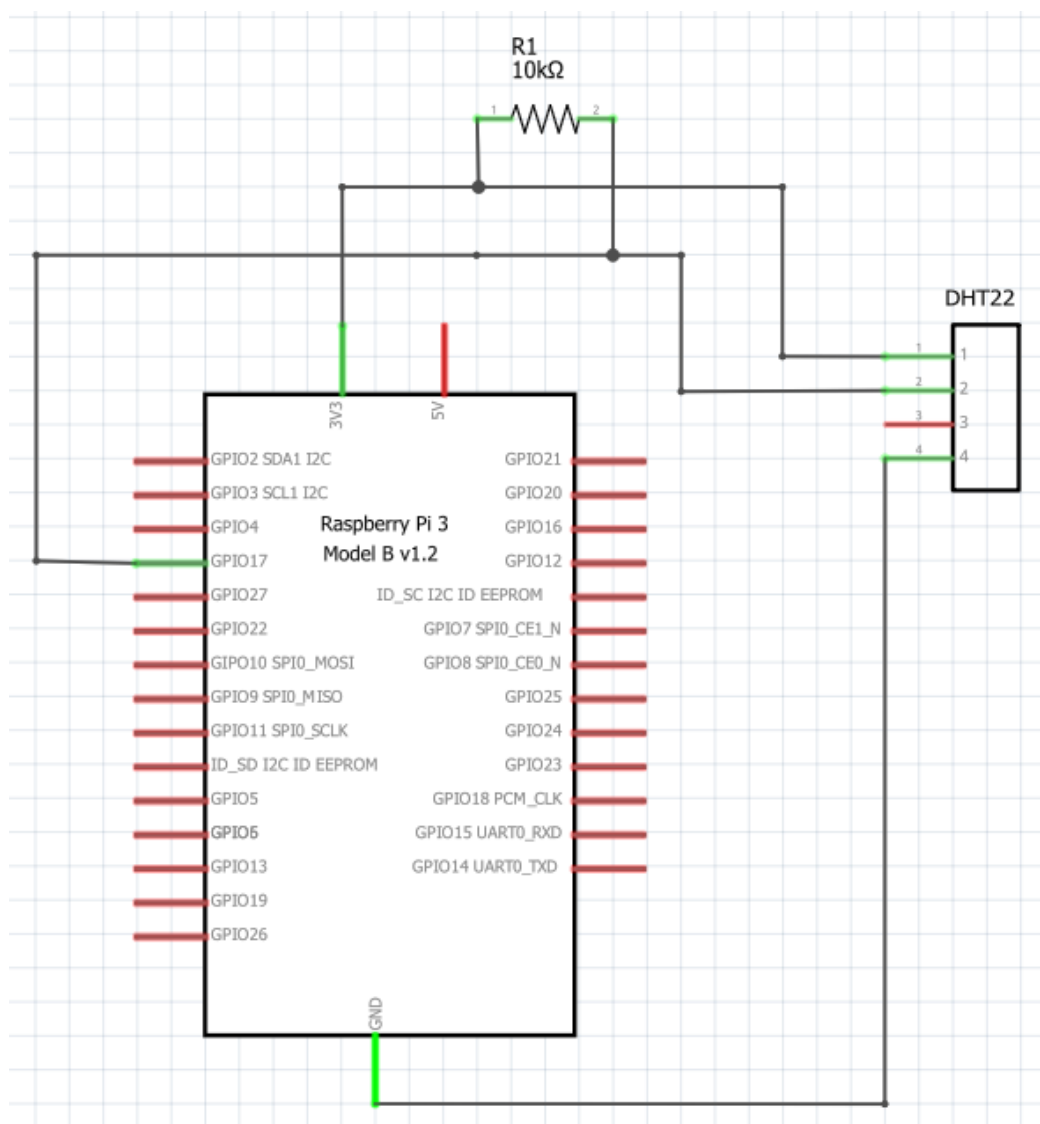
3. Esquemas de los circuitos y conexión de sensores.

Ahora que ya tenemos la RPi funcionando correctamente con conexión Wi-Fi, podemos acceder y jugar a nuestro antojo con él, ya estamos en disposición de dar un paso más allá.

En este apartado vamos a dar a conocer todos los esquemas/circuitos electrónicos, es decir, la conexión de los sensores a utilizar con la RPi, los pasos necesarios para instalar las librerías de cada sensor y también se muestran los scripts (código en Python) utilizados para capturar los datos de los sensores.

Para simplificar y mejorar el entendimiento del proyecto en este apartado no se incluye nada relacionado con la nube. Los scripts que se muestran, solo capturan los datos de los sensores y se muestran en la terminal de PuTTY. En los apartados siguientes se explica cómo relacionarlo con la nube y crear un web con gráficos que muestre estos datos.

3.1. Conexión e instalación del sensor de Temperatura y Humedad DHT22



Esquema 3.1. Circuito de conexión del sensor de Humedad y Temperatura DHT22. Fuente: Pantallazo PC

Como el propio nombre indica se trata de la conexión e instalación de la librería de un sensor de temperatura y humedad DHT22. En el esquema 1, se puede ver cómo de sencilla es la conexión de este sensor.

Una vez se ha realizado la conexión tal y como se indica en el esquema, pasamos a instalar la librería del sensor con los siguientes comandos:

a. Clonamos el repositorio (https://github.com/adafruit/Adafruit_Python_DHT.git) del sensor DHT22 Python:

```
pi@raspberrypi:~$ git clone
https://github.com/adafruit/Adafruit_Python_DHT.git
```

b. Entramos en el repositorio clonado e instalamos la librería:

```
pi@raspberrypi:~$ cd Adafruit_Python_DHT/
pi@raspberrypi:~$ sudo python3 setup.py install
```

Ahora ya estamos en disposición de poder utilizar el sensor DHT22. Para probar su funcionamiento daremos los siguientes pasos:

a. Crear un archivo Python con el editor de texto que deseemos y pegamos el código que se muestra a continuación:

```
import Adafruit_DHT

DHT_SENSOR = Adafruit_DHT.DHT22
DHT_PIN = 17

while True:
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)

    if humidity is not None and temperature is not None:
        print("Temp={0:0.1f}*C  Humidity={1:0.1f}%".format(temperature,
humidity))
    else:
        print("Failed to retrieve data from humidity sensor")
```

Script 3.1. Humidity.py

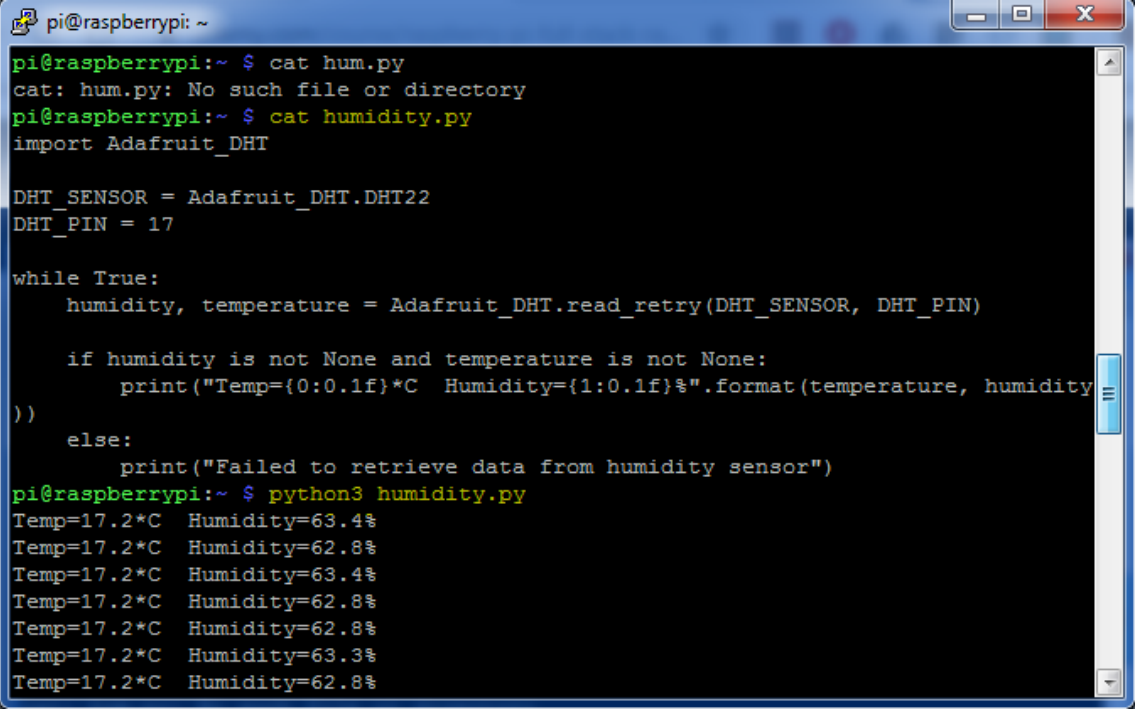
b. Subimos el archivo creado a la RPi por medio de FileZilla.

Existe la posibilidad de utilizar el editor de texto VIM de la propia RPi, pero para simplificar los pasos se da a conocer este método.

c. Ejecutamos el código:

```
pi@raspberrypi:~$ python3 humidity.py
```

El resultado debe ser parecido al que se muestra en la siguiente imagen:



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ cat hum.py  
cat: hum.py: No such file or directory  
pi@raspberrypi:~ $ cat humidity.py  
import Adafruit_DHT  
  
DHT_SENSOR = Adafruit_DHT.DHT22  
DHT_PIN = 17  
  
while True:  
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)  
  
    if humidity is not None and temperature is not None:  
        print("Temp={0:0.1f}*C Humidity={1:0.1f}%".format(temperature, humidity  
    ))  
    else:  
        print("Failed to retrieve data from humidity sensor")  
pi@raspberrypi:~ $ python3 humidity.py  
Temp=17.2*C Humidity=63.4%  
Temp=17.2*C Humidity=62.8%  
Temp=17.2*C Humidity=63.4%  
Temp=17.2*C Humidity=62.8%  
Temp=17.2*C Humidity=62.8%  
Temp=17.2*C Humidity=63.3%  
Temp=17.2*C Humidity=62.8%
```

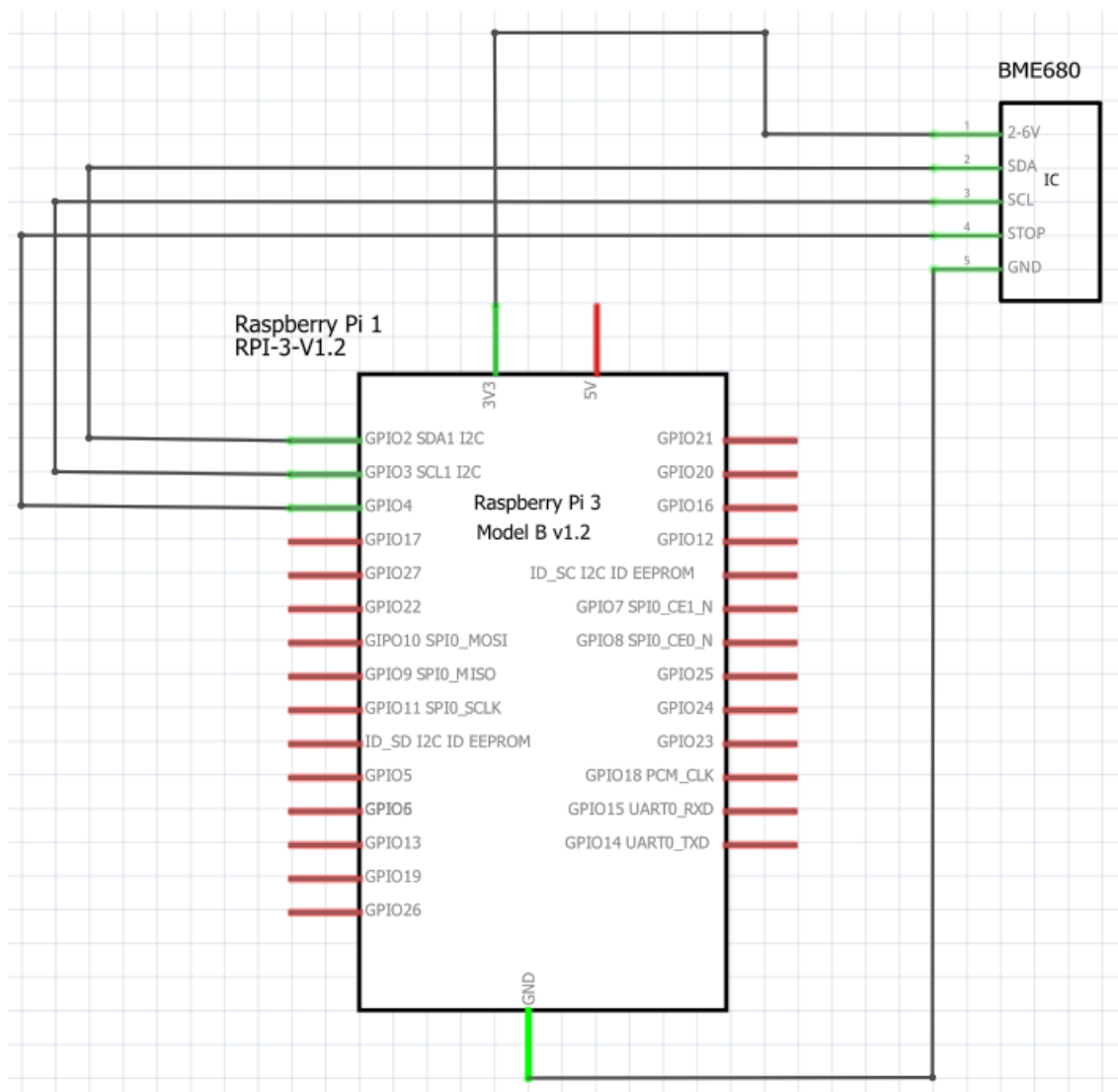
Figura 3.1. Resultados del sensor DHT22. Fuente: Pantallazo PC

El comando `cat humidity.py` muestra únicamente la información que hay dentro de un archivo Python, txt o cualquier otro tipo de archivo.

Se puede ver en la Figura 3.1, al ejecutar la sentencia `Python3 humidity.py`, se van mostrando los valores de Temperatura y Humedad.

3.2. Conexión e instalación del sensor de BME680

El sensor BME680, es un sensor que permite medir la temperatura (°C), la humedad (%RH), la calidad del aire (Ohms) y la presión atmosférica del ambiente (hPa). Este sensor resulta bastante cómodo ya que con un único dispositivo de un tamaño reducido podemos medir 4 datos del medio ambiente.



Esquema 3.2. Circuito de conexión del sensor BME680. Fuente: elaboración propia

En la Esquema 2, se muestra el circuito de conexión. Una vez se monta el circuito tal y como se muestra en el esquema se procede a encender la RPi e instalar e instalar las librerías y realizar las pruebas necesarias para su correcto funcionamiento. Para ello seguiremos los siguientes pasos:

a. Habilitamos el protocolo de comunicación I2C con los siguientes comandos:

```
pi@raspberrypi:~$ sudo su
root@raspberrypi:/home/pi# curl https://get.pimoroni.com/i2c |
bash
```

b. Clonamos el repositorio del sensor BME680 e instalamos la librería para Python 3:

```
root@raspberrypi:/home/pi# git clone
```

```
https://github.com/pimoroni/bme680-python.git
```

```
root@raspberrypi:/home/pi# sudo python3 setup.py install
```

c. En el caso de que nos de error por no disponer de un módulo llamado “smbus”. Proceder a instalar con la siguiente instrucción:

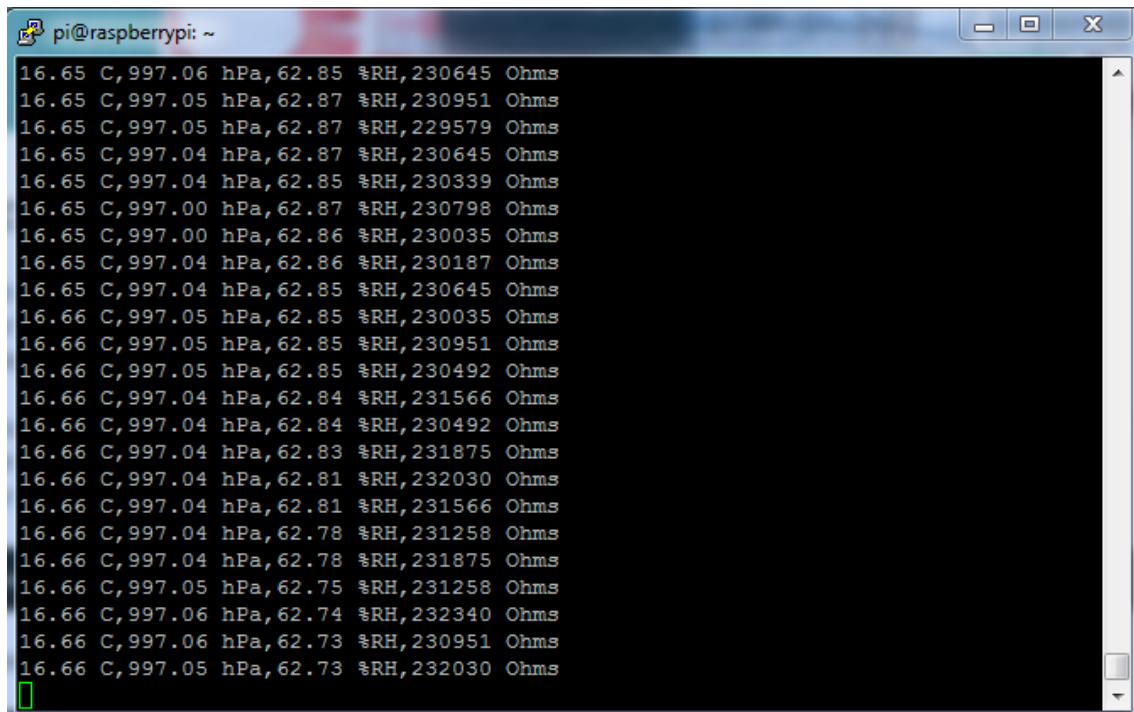
```
root@raspberrypi:/home/pi# sudo apt-get install python3-smbus
```

Ahora ya estamos en disposición de poder utilizar el sensor BME680. Para probar su funcionamiento utilizaremos uno de los ejemplos que nos proporciona la propia librería, para ello ejecutamos los siguientes comandos:

```
root@raspberrypi:/home/pi/bme680# cd examples/
```

```
root@raspberrypi:/home/pi/bme680/examples# python read-all.py
```

Al ejecutar el script read-all.py, lo que muestra es lo siguiente:



```
pi@raspberrypi: ~
16.65 C,997.06 hPa,62.85 %RH,230645 Ohms
16.65 C,997.05 hPa,62.87 %RH,230951 Ohms
16.65 C,997.05 hPa,62.87 %RH,229579 Ohms
16.65 C,997.04 hPa,62.87 %RH,230645 Ohms
16.65 C,997.04 hPa,62.85 %RH,230339 Ohms
16.65 C,997.00 hPa,62.87 %RH,230798 Ohms
16.65 C,997.00 hPa,62.86 %RH,230035 Ohms
16.65 C,997.04 hPa,62.86 %RH,230187 Ohms
16.65 C,997.04 hPa,62.85 %RH,230645 Ohms
16.66 C,997.05 hPa,62.85 %RH,230035 Ohms
16.66 C,997.05 hPa,62.85 %RH,230951 Ohms
16.66 C,997.05 hPa,62.85 %RH,230492 Ohms
16.66 C,997.04 hPa,62.84 %RH,231566 Ohms
16.66 C,997.04 hPa,62.84 %RH,230492 Ohms
16.66 C,997.04 hPa,62.83 %RH,231875 Ohms
16.66 C,997.04 hPa,62.81 %RH,232030 Ohms
16.66 C,997.04 hPa,62.81 %RH,231566 Ohms
16.66 C,997.04 hPa,62.78 %RH,231258 Ohms
16.66 C,997.04 hPa,62.78 %RH,231875 Ohms
16.66 C,997.05 hPa,62.75 %RH,231258 Ohms
16.66 C,997.06 hPa,62.74 %RH,232340 Ohms
16.66 C,997.06 hPa,62.73 %RH,230951 Ohms
16.66 C,997.05 hPa,62.73 %RH,232030 Ohms
```

Figura 3.2. Resultados del Script read-all.py del sensor BME680. Fuente: Pantallazo PC

En la Figura 3.2, podemos ver de izquierda a derecha los datos de los diferentes sensores. Todos resultan fáciles de interpretar excepto el último dato que es la calidad del aire en Ohms.

Los datos de calidad de aire en Ohms, es un dato que comienza desde el más bajo y va subiendo hasta llegar a estabilarse después de unos minutos (si es la primera vez que se utiliza el sensor, conviene dejarlo funcionando entre 15 minutos – 30 minutos). Cuanto más alto son los Ohms, mayor calidad del aire.

Si ejecutamos la sentencia `cat read-all.py`, se puede ver el código Python y tratar de entender la lógica que hay detrás.

3.3. Contratiempos sensores.

Los circuitos correspondientes con el sensor de intensidad de luz (Luxometro) y el medidor de energía con protocolo de comunicación modbus, no se muestran en este apartado porque nos hemos encontrado con contratiempos realizando esta fase experimental.

Los contratiempos encontrados con estos dos sensores son los siguientes:

Medidor de energía monofásico con protocolo de comunicación Modbus. Al realizar mal el cableado eléctrico, se ha provocado un cortocircuito dejando el aparato completamente inutilizado. Al venir el producto desde china, tanto el tiempo de envío como el coste hacían poco viable comprar otro para continuar con los experimentos.

Sensor de intensidad de luz. Ha llegado defectuoso de fábrica, se han realizado las conexiones con la Raspberry Pi según las instrucciones, pero no se ha conseguido comunicación a través del bus I2C.

4. Preparación y comunicación con la nube.

De todos los sensores mostrados en el apartado anterior, la idea es crear una única plataforma que visualice por medio de gráficos todos estos datos capturadas por los sensores. Pero a efectos prácticos solo se explica lo más detalladamente posible cómo crear una plataforma que visualice únicamente los datos de humedad y temperatura.

Es mucho más fácil de entender y asimilar si se muestra el flujo del proceso de creación del concepto de IoT si nos centramos únicamente en implementar unos pocos datos sin llegar a complicarnos demasiado. Una vez se explica cómo crear la plataforma se procede a mostrar a través de GitHub el código completo de esta parte simplificada y de la parte completa con todos los sensores.

4.1. Arquitectura básica del concepto IoT.

En la Figura 4.1, se muestra la arquitectura que se va a explicar, como se puede ver es simple y se utilizan los servicios esenciales de la nube para crear el concepto de IoT.

En la parte de Hardware se puede ver la RPi 3 B+ en la cual se han añadido dentro certificaciones IoT de la nube y AWS SDK para Python para mejorar la seguridad y establecer la comunicación con la nube respectivamente. Más adelante se explica cómo crear y poner dentro de la RPi estas certificaciones.

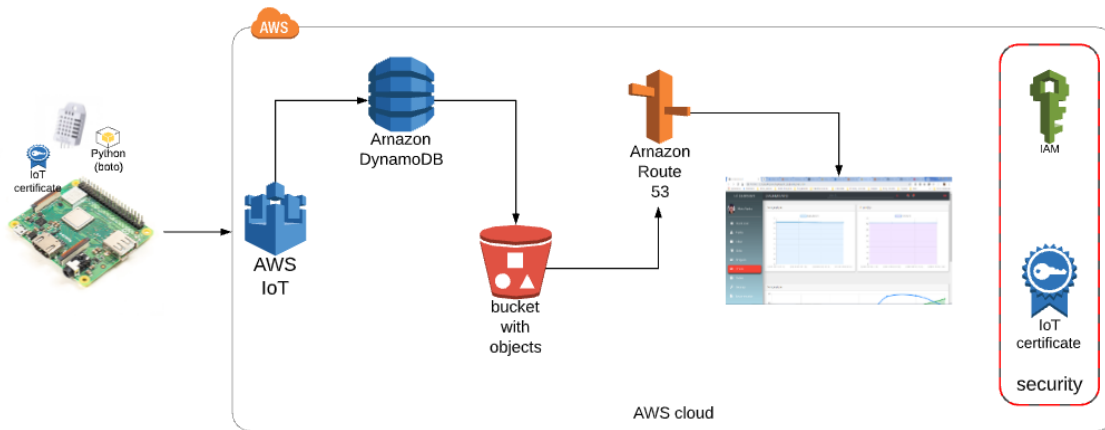


Figura 4.1. Arquitectura IoT con un sensor DHT22. Fuente: elaboración propia

En la parte correspondiente a la nube de AWS tendemos los siguientes servicios realizando las siguientes funciones específicas:

- **AWS IoT**, como ya se sabe, se trata de un conjunto de muchos servicios que ofrece AWS, relacionados con el Internet de la Cosas. En este caso se utiliza para crear el canal de comunicación entre la RPi y la nube, crear los certificados de seguridad, recibir los datos de los sensores por MQTT y enviar los datos a una base de datos no relacional llamada DynamoDB.
- **DynamoDB**, recibe y almacena los datos de los sensores de una forma que sea fácil de acceder por cualquier lenguaje de programación, como puede ser Python, Java, Node.js o JavaScript. Con esta arquitectura accederemos a los datos de DynamoDB con JavaScript y un AWS SDK para JavaScript.
- **S3 (Bucket with objects)**, aquí se almacenará todo el código relacionado con la página web, es decir, archivos CSS, HTML y JavaScript, con una estructura de carpetas adecuada. Con solo esto ya tendríamos la web y podríamos acceder a su visualización con un dominio por defecto de AWS.
- **Route53**, nos permite comprar un dominio de pago y de nuestro agrado, permitiendo que sea enlazado en el cubo (Bucket) de S3, donde tengamos almacenados los archivos de nuestra web. Por lo que bastará con poner un nombre de dominio fácil para poder acceder a la visualización de los datos.

En los siguientes apartados se explica cómo configurar cada servicio de la nube de AWS.

4.2. Configuración de AWS IoT.

En este apartado se explica cómo configurar el servicio de la nube relacionado con IoT, este servicio se llama AWS IoT y hay que decir que la manera en la que se utiliza este servicio entra dentro de la capa gratuita (https://aws.amazon.com/es/free/?nc1=h_ls) de AWS por lo que no tiene “ningún coste”.

Recordar que en este paso ya se ha instalado las librerías necesarias para la comunicación con AWS, por lo que en este apartado solo se centra en lo siguiente:

- Registrar un objeto, es decir, nuestra RPi 3 B+
- Obtención de los certificados de seguridad y guardarlos en la RPi, para una comunicación segura entre nube y objeto (RPi 3 B+)

Suponiendo que estamos registrados en AWS vamos a servicios y entramos en **AWS IoT**. En la parte izquierda veremos un menú, hacemos clic en **Administración** y después en **objetos**.

- **Servicios >> AWS IoT >> Administración >> Objetos**

Una vez en objetos, si es la primera vez que creamos un objeto se nos aparecerá una imagen como la que se muestra en la Figura 4.2. Una vez ahí le damos a **Registrar un objeto**.

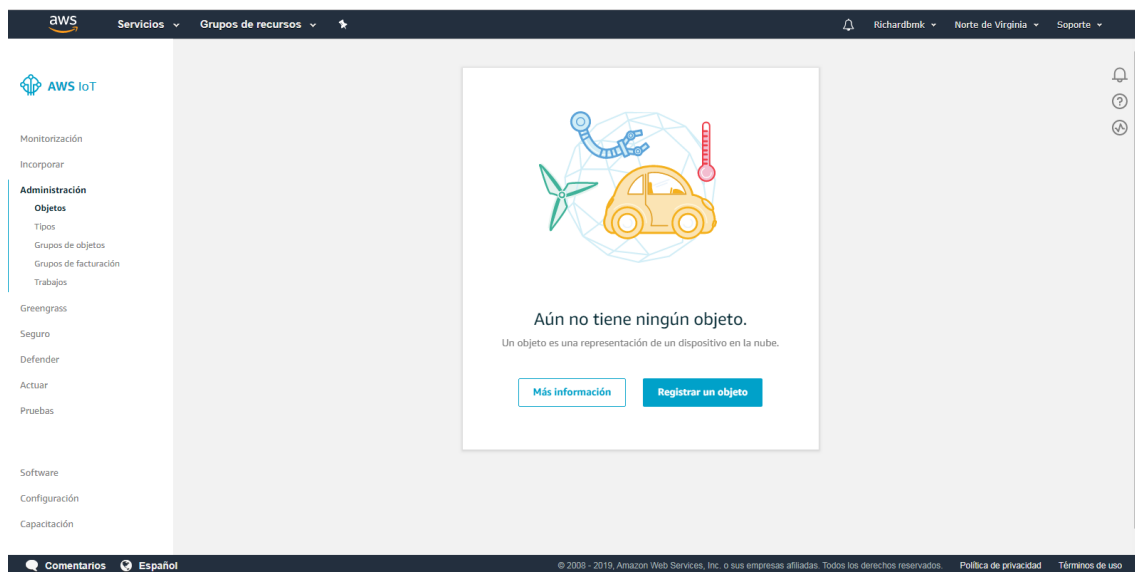


Figura 4.2. Ventana de inicio de AWS. Fuente: Pantallazo PC

Una vez le damos a Registrar objeto, se nos abrirá una ventana como la que se muestra en la figura 4.3, en la cual se nos dan dos opciones crear un objeto o lotes de objetos. Elegimos el caso que nos ocupa, que es **crear un solo objeto**.

- **Registrar objeto >> crear un solo objeto**

Creación de objetos de AWS IoT

Un objeto de IoT es la representación y el registro de su dispositivo físico en la nube. Cualquier dispositivo físico necesita un registro de objeto para poder funcionar con AWS IoT. [Más información.](#)

Registrar un solo objeto de AWS IoT
Cree un objeto en el registro. Crear un solo objeto

Registrar por lotes muchos objetos de AWS IoT
Cree objetos en el registro para un gran número de dispositivos que ya utilicen AWS IoT o registre los dispositivos para que se puedan conectar a AWS IoT. Crear muchos objetos

Cancelar Crear un solo objeto

Figura 4.3. Elección de cómo registrar objetos. Fuente: Pantallazo PC

Una vez le hayamos dado a esa opción nos encontraremos con la figura 4.4, en el cual se nos pide el nombre que deseamos darle al objeto. Ponemos el nombre, dejamos todo lo demás por defecto y le damos a siguiente.

CREAR UN OBJETO PASO 1/3

Añadir su dispositivo al registro de objetos

Este paso crea una entrada en el registro de objetos y una sombra de objeto para el dispositivo.

Nombre

Aplicar un tipo a este objeto
El uso de tipos de objetos simplifica la administración de dispositivos al proporcionar los mismos datos de registro para los objetos que comparten un tipo. Los tipos proporcionan a los objetos un conjunto común de atributos, que describen la identidad y las funciones del dispositivo, así como una descripción.

Tipo de objeto
No se ha seleccionado ningún tipo Crear un tipo

Añadir este objeto a un grupo
Añadir el objeto a un grupo le permite administrar los dispositivos de forma remota mediante trabajos.

Grupo de objetos
Grupos / Crear grupo Cambiar

Figura 4.4. Paso 1/3. Nombre del objeto. Fuente: Pantallazo PC

La siguiente imagen con la que nos encontraremos, es la imagen de la Figura 4.5, en la que se nos pide crear un certificado para nuestro objeto. Como se puede ver en la misma figura hay unas cuantas opciones, escogemos la primera que es la opción más fácil. Le damos a **Crear certificado**.

CREAR UN OBJETO

Añadir un certificado para el objeto

PASO 2/3

Los certificados se utilizan para autenticar la conexión del dispositivo con AWS IoT.

Creación de un certificado con un clic (recomendado)
Se generará un certificado, una clave pública y una clave privada mediante la entidad de certificación de AWS IoT.

Crear un certificado

Crear con CSR
Cargue su propia solicitud de firma de certificado (CSR) basada en su propia clave privada.

Crear con CSR

Usar mi certificado
Registre su certificado de CA y use sus propios certificados en todos los dispositivos que desee.

Introducción

Omitir certificado y crear objeto
Tendrá que añadir un certificado al objeto más adelante para que su dispositivo se pueda conectar a AWS IoT.

Crear un objeto sin certificado

Figura 4.5. Paso 2/3. Creación del certificado. Fuente: Pantallazo PC

Una vez le demos a la opción **Crear certificado**, se nos aparecerá la imagen de la Figura 4.6, donde podremos descargarnos los certificados creados. Descargar los tres primeros certificados.

Hay un cuarto archivo que es necesario descargarse, este es el certificado raíz (CA raíz), al darle a descargar nos lleva a otra página de AWS. No es exactamente una descarga, para obtener este archivo es necesario seguir unos pasos diferentes:

1. Le damos a la pestaña **descargar** y nos lleva a la siguiente página o parecida:

<https://docs.aws.amazon.com/iot/latest/developerguide/server-authentication.html#server-authentication-certs>

2. Dentro de la web que nos ha llevado al darle clic, buscar con Ctrl + B la frase **“Amazon Trust Services Endpoints (preferred)”**, en la Figura 4.7, se muestra una imagen. Una vez ahí veremos cuatro opciones, le damos clic a la primera que nos llevará a otro archivo tal y como se muestra en la figura 4.8.

3. Nos encontraremos con un montón de texto sin sentido, igual que se muestra en la figura X. Copiamos y pegamos este texto en cualquier editor de texto como Notepad.

4. Una vez creado el archivo, cambiamos el nombre y la extensión por uno con la terminación pem. Ejemplo: **CA-root.txt >>> CA-root.pem**

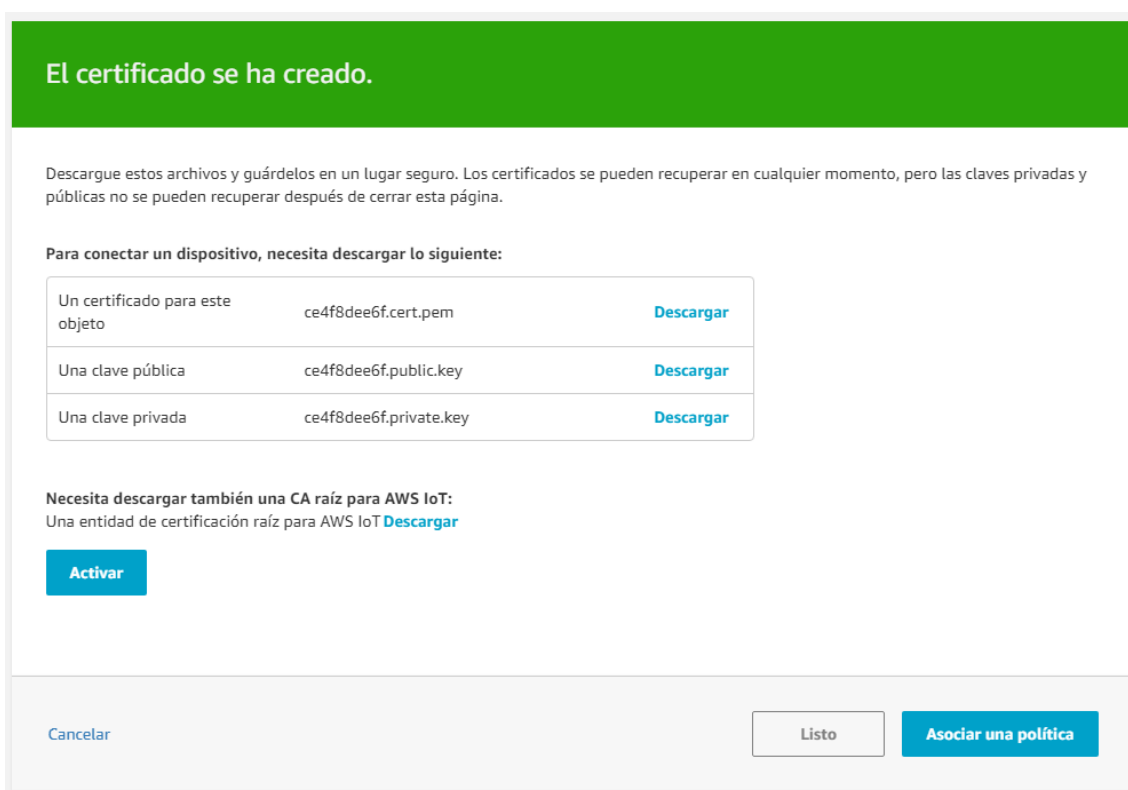


Figura 4.6. Paso 2/3. Certificado creado. Fuente: Pantallazo PC

CA Certificates for Server Authentication

Depending on which type of data endpoint you are using and which cipher suite you have negotiated, AWS IoT server authentication certificates are signed by one of the following root CA certificates:

VeriSign Endpoints (legacy)

- RSA 2048 bit key: [VeriSign Class 3 Public Primary G5 root CA certificate](#)

Amazon Trust Services Endpoints (preferred)

- RSA 2048 bit key: [Amazon Root CA 1](#).
- RSA 4096 bit key: Amazon Root CA 2. Reserved for future use.
- ECC 256 bit key: [Amazon Root CA 3](#).
- ECC 384 bit key: Amazon Root CA 4. Reserved for future use.

These certificates are all cross-signed by the [Starfield Root CA Certificate](#). All new AWS IoT Core regions, beginning with the May 9, 2018 launch of AWS IoT Core in the Asia Pacific (Mumbai) Region, serve only ATS certificates.

Figura 4.7. Descargar el archivo CA raíz. Fuente: Pantallazo PC

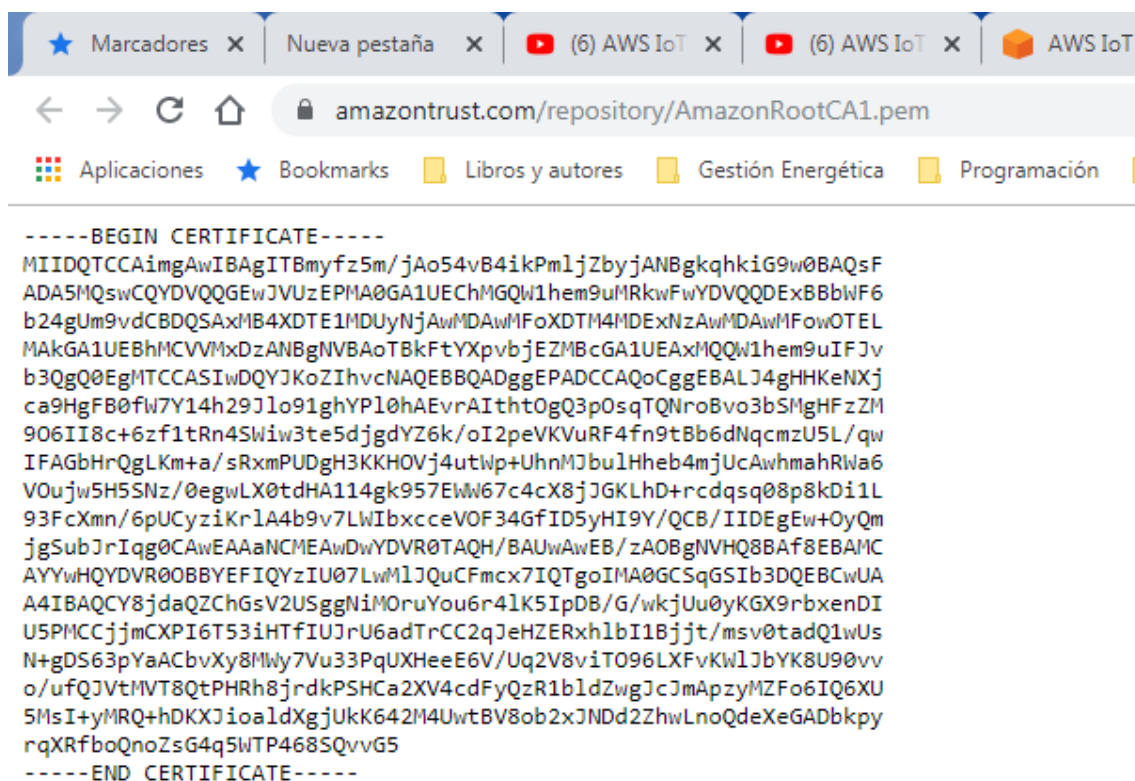


Figura 4.8. Contenido del archivo CA raíz. Fuente: Pantallazo PC

Una vez ya se han descargado estos 4 archivos, ya podemos darle al botón de **activar** de la Figura 4.6 y después al botón de **asociar una política**. Como en estos momentos no tenemos ninguna política, le daremos a registrar objeto y en los siguientes pasos, explicaremos que es una política en AWS y crearemos una como las que se muestran en la figura 4.8.

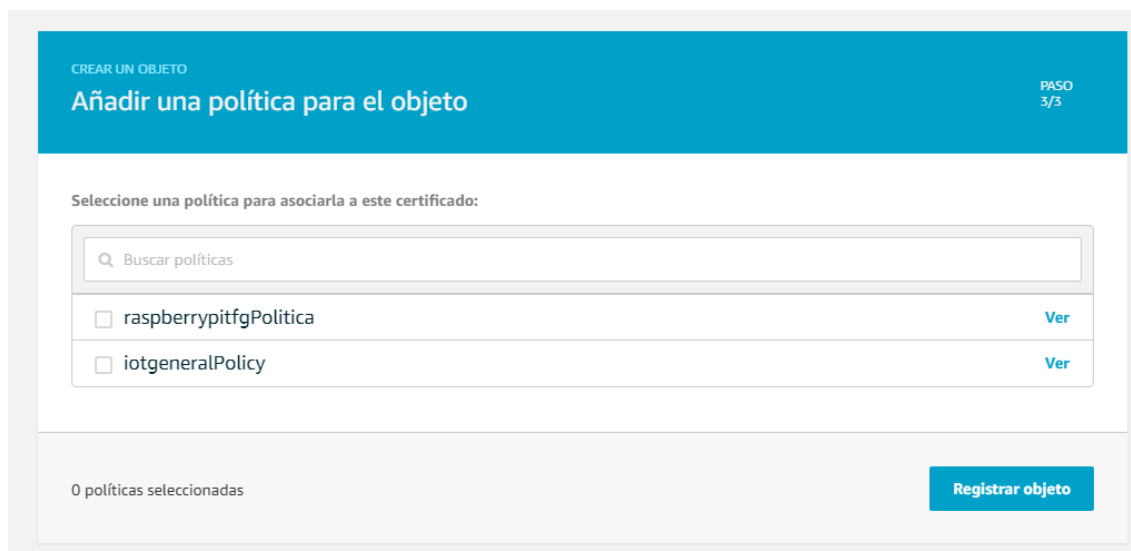


Figura 4.8. Paso 3/3. Selección de una política. Fuente: Pantallazo PC

Los certificados descargados son importantes y serán necesarios más adelante, por lo que es necesario guardarlos en una carpeta en nuestro ordenador de momento.

Creación de Políticas para AWS IoT.

Las políticas, en general sirven para definir los privilegios que le daremos a un determinado servicio en AWS. Es un documento en formato JSON, que en el cual se pone una serie de instrucciones que permite o deniegan una acción en concreto, como, por ejemplo: escribir, leer, eliminar, etc. Ejemplo de una y la que utilizaremos para AWS IoT, es la siguiente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:*",
      "Resource": "*"
    }
  ]
}
```

Script 4.1. JSON Script para política de AWS IoT.

Esta política tal y como está escrita, nos permite acceso para realizar cualquier acción en el servicio de AWS IoT. El asterisco significa “full access”.

No es necesario hacer este script, AWS ya lo hace por nosotros a base de hacer clics y escribir muy poco.

Para crear la política de AWS IoT y asociarla al objeto que acabamos de crear, es necesario realizar los siguientes pasos:

Ir al servicio de **AWS IoT**, a la izquierda hacer clic en la opción **Seguro** y después a la opción **Políticas**. Una vez en políticas le damos a la opción Crear. Si nunca hemos creado una política se nos mostrará una imagen diferente a la que se muestra en la Figura 4.9, pero igual de fácil e intuitiva.

- AWS IoT >> Seguro >> Políticas >> Crear

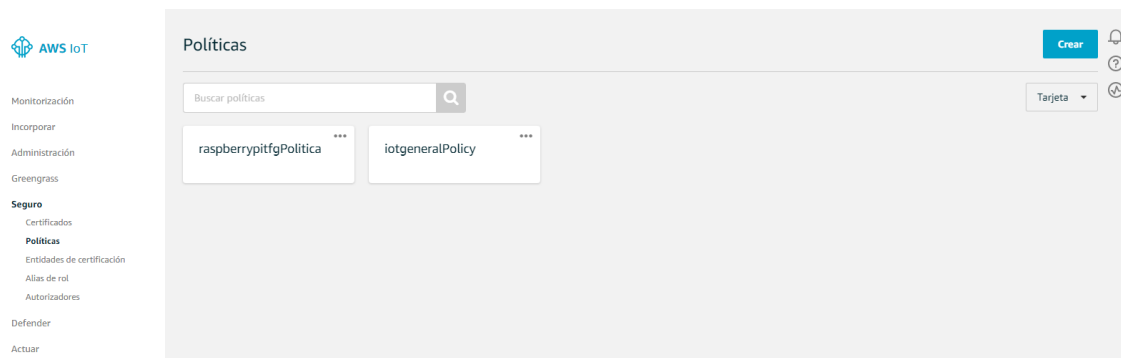


Figura 4.9. Ventana de creación de políticas. Fuente: Pantallazo PC

Una vez le damos a crear, nos encontraremos con la Figura 4.10, la cual deberemos rellenar con los siguientes datos:

Nombre: nombre_qDeseemos

Acción: iot: *

ARN de recurso: *

Figura 4.10. Configuración política en AWS IoT. Fuente: Pantallazo PC

Una vez se ha rellenado todo le damos a Crear. Y con esto ya lo tendríamos. Si entramos dentro de la política creada podremos ver el archivo JSON, que acabamos de crear con instrucciones iguales o parecidas al script 1.

Ahora debemos asociar el certificado de nuestro objeto creado con la política que acabamos de crear. Para ello vamos al **objeto creado**, una vez dentro pulsamos **seguridad**, opción que se encuentra a la izquierda. Nos encontraremos con el **certificado creado**, hacemos clic en ese certificado, una vez dentro del certificado, hacemos clic en **acciones** y finalmente pulsamos en **asociar política**, seleccionamos la **política creada** y pulsamos **asociar**.

- **objetoCreado >> Seguridad >> CertificadoCreado >> Acciones >> Asociar politica >> SelecciónDePolitica >> Asociar**

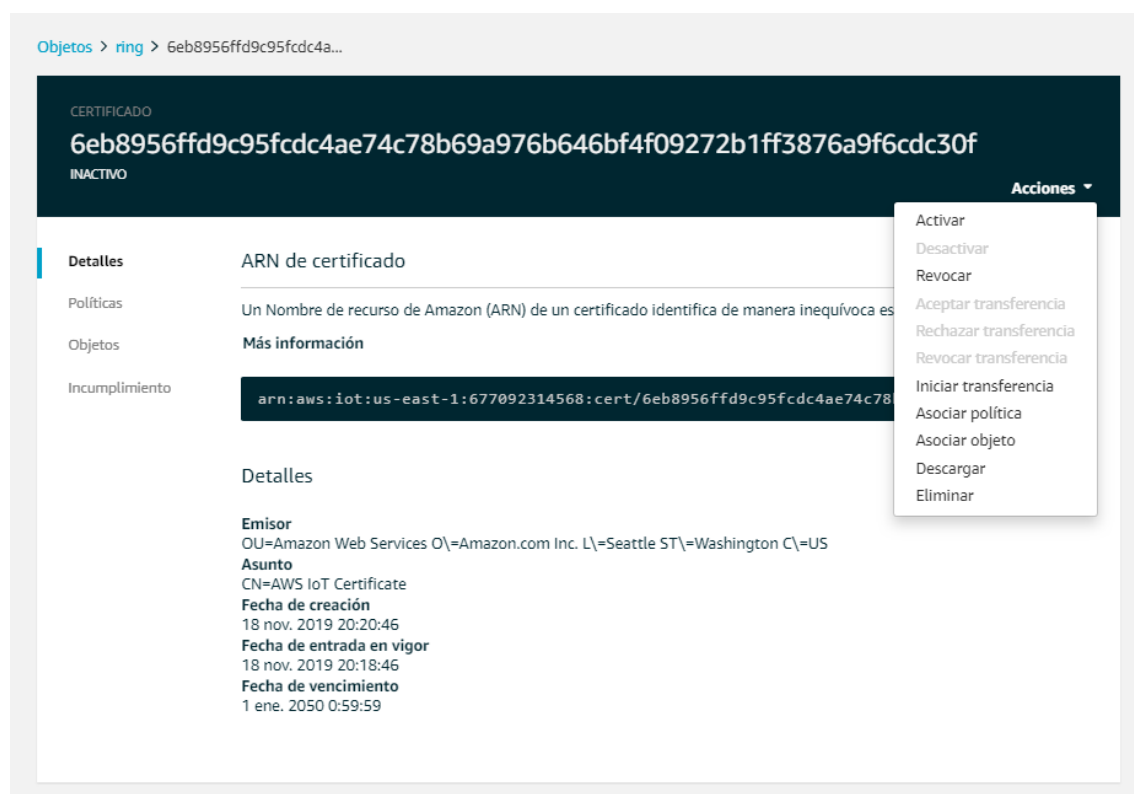


Figura 4.11. Asociar política al objeto IoT. Fuente: Pantallazo PC

Con esto hemos acabado la configuración del objeto en AWS IoT, ahora vamos a probar su funcionamiento metiendo los certificados creado en la RPI y ejecutando un script que envíe los datos únicamente a AWS IoT.

En entramos en la RPI, como ya se ha explicado anteriormente y creamos una carpeta con el siguiente nombre **awsiot** o cualquiera de nuestro gusto. Dentro de esta carpeta creamos otra carpeta con el nombre **certificados** donde pondremos los certificados creados.

Los certificados creados, suelen venir por defecto con nombres raros como los que se muestra en la figura 4.12, se pueden cambiar si lo deseamos.

2a1aa524d5-certificate.pem.crt	13/07/2019 21:18	Certificado de seg...	2 KB
2a1aa524d5-private.pem.key	13/07/2019 21:18	Archivo KEY	2 KB
2a1aa524d5-public.pem.key	13/07/2019 21:18	Archivo KEY	1 KB
root-CA.pem	13/07/2019 21:19	Archivo PEM	2 KB

Figura 4.12. Certificados del objeto Creado en AWS IoT. Fuente: Pantallazo PC

Recordar con FileZilla resulta fácil añadir cualquier archivo en la RPi. Si optamos por esta opción podremos arrastrar los certificados del ordenador a la carpeta certs creada de nuestra RPi, tal y como se muestra en la Figura 4.13.

En la Figura 4.13, podemos ver que también tenemos un archivo con el nombre realdataPub14.py. Este archivo contiene el código que se conecta a AWS IoT, captura los datos sensados gracias a la librería de Adafruit_DHT y enviar los datos de manera segura con un protocolo MQTT.

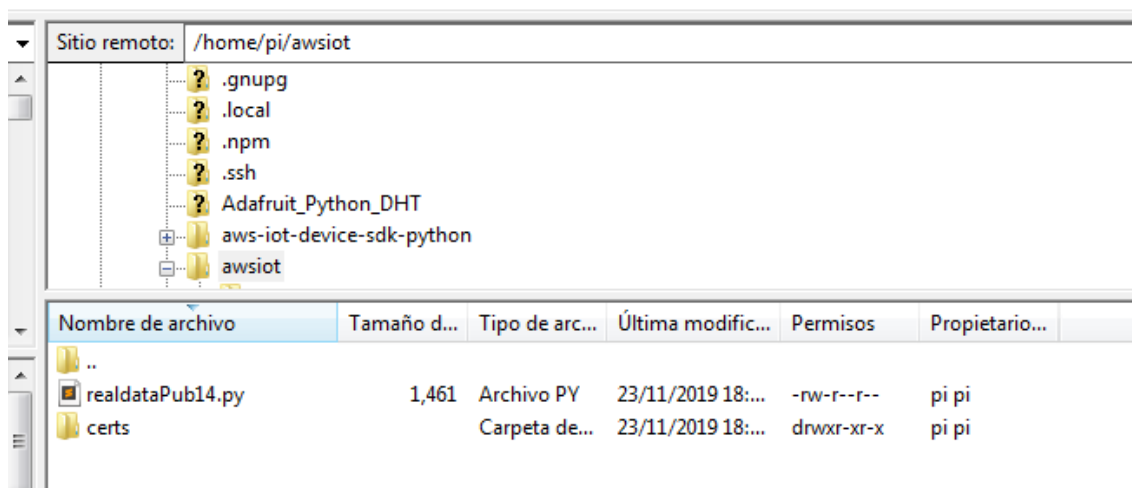


Figura 4.13. La RPi a través de FileZilla. Fuente: Pantallazo PC

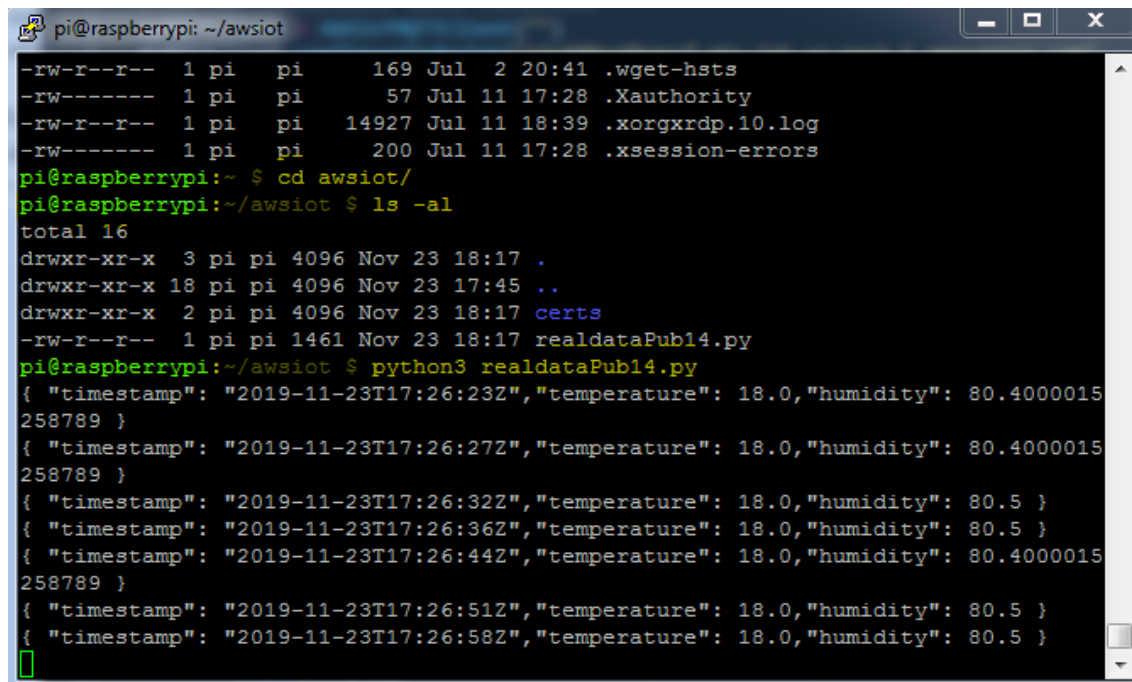


Figura 4.13. Ejecución del archivo realdataPub14.py. Fuente: Pantallazo PC

En la Figura 4.13, podemos ver unos de los resultados de ejecutar el archivo realdataPub14.py. Si nos fijamos podemos observar que son los datos del sensor en formato JSON. Estos mismos datos se están enviando a la nube. Más adelante veremos cómo confirmar que se están

enviando los datos a la nube a través de AWS IoT, pero primero intentaremos explicar el contenido del archivo de tal manera que se entienda.


```
1) import Adafruit_DHT
2) from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient
3) from time import sleep
4) from datetime import date, datetime

5) # AWS IoT certificate based connection
6) myMQTTClient = AWSIoTMQTTClient("")
7) myMQTTClient.configureEndpoint("a3v600sq9kqevf-ats.iot.us-east-1.amazonaws.com",
8883)
8) myMQTTClient.configureCredentials("certs/x590_root.crt", "certs/d68c299b76-
private.pem.key", "certs/d68c299b76-certificate.pem.crt")
9) myMQTTClient.configureOfflinePublishQueueing(-1) # Infinite offline Publish
queueing
10) myMQTTClient.configureDrainingFrequency(2) # Draining: 2 Hz
11) myMQTTClient.configureConnectDisconnectTimeout(10) # 10 sec
12) myMQTTClient.configureMQTTOperationTimeout(5) # 5 sec

13) # connect and publish
14) myMQTTClient.connect()
15) myMQTTClient.publish("thing01/info", "connected", 0)

16) sensor = Adafruit_DHT.DHT22
17) pin = 17 # GPIO 17

18) # loop and publish sensor reading
19) while 1:
20)     now = datetime.utcnow()
21)     now_str = now.strftime('%Y-%m-%dT%H:%M:%SZ') # e.g. 2016-04-18T06:12:25.877Z
22)     humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
23)     if humidity is not None and temperature is not None:
24)         payload = '{ "timestamp": "' + now_str + '", "temperature": ' +
str(temperature) + ', "humidity": ' + str(humidity) + ' }'
25)         print(payload)
26)         myMQTTClient.publish("thing01/data", payload, 0)
27)         sleep(4)
28)     else:
29)         print(".")
30)         sleep(1)
```

Script 4.2. contenido realdataPub14.py.

Explicación de Script 4.2.

En el script 4.2, podemos ver el código completo del archivo `realdataPub.py`. A continuación, se explica en términos generales el código:

Sentencias 1 a 4: son las sentencias que importan las librerías necesarias que hemos instalado anteriormente y otras librerías que vienen por defecto.

La línea 2, se encarga de importar todo lo que necesitamos para conectarnos a AWS IoT.

La línea 3, importa `sleep` que sirve para esperar el tiempo que deseemos antes de volver a ejecutar una el archivo, de la librería `time`.

La línea 4, importamos la librería `datetime` y de esa librería extraemos las funciones `date` y `datetime`, necesarios para adquirir los datos del tiempo en el cual se han registrado los datos de temperatura y humedad.

Sentencias 6 a 12: son las sentencias necesarias para configurar y conectar el archivo Python con la nube, es decir con el servicio de AWS IoT.

La línea 7 `configureEndpoint`, permite a AWS saber dónde se encuentra nuestro objeto, el cual está definido por una ruta.

Línea 8 `configureCredentials`, decimos a AWS IoT la ruta dónde encontrar las credenciales para permitir o no la comunicación con la nube.

Sentencias 13 a 15: permiten la conexión y publicación de los datos en AWS IoT.

Para entender mejor la librería de `AWSIoTPythonSDK` se recomienda acudir a las librerías oficiales:

- <https://pypi.org/project/AWSIoTPythonSDK/1.0.0/>
- <https://s3.amazonaws.com/aws-iot-device-sdk-python-docs/sphinx/html/index.html>

Sentencias 19 a 30: bucle infinito que lee los datos sensados y los publica después a través de un tópico en AWS IoT.

Para probar que funciona correctamente nos dirigimos a AWS IoT y después vamos a pruebas.

Ponemos el nombre del tópico con que se publican los datos, en este caso "thing01/data". Y le damos a publicar temas. Figura 4.14.

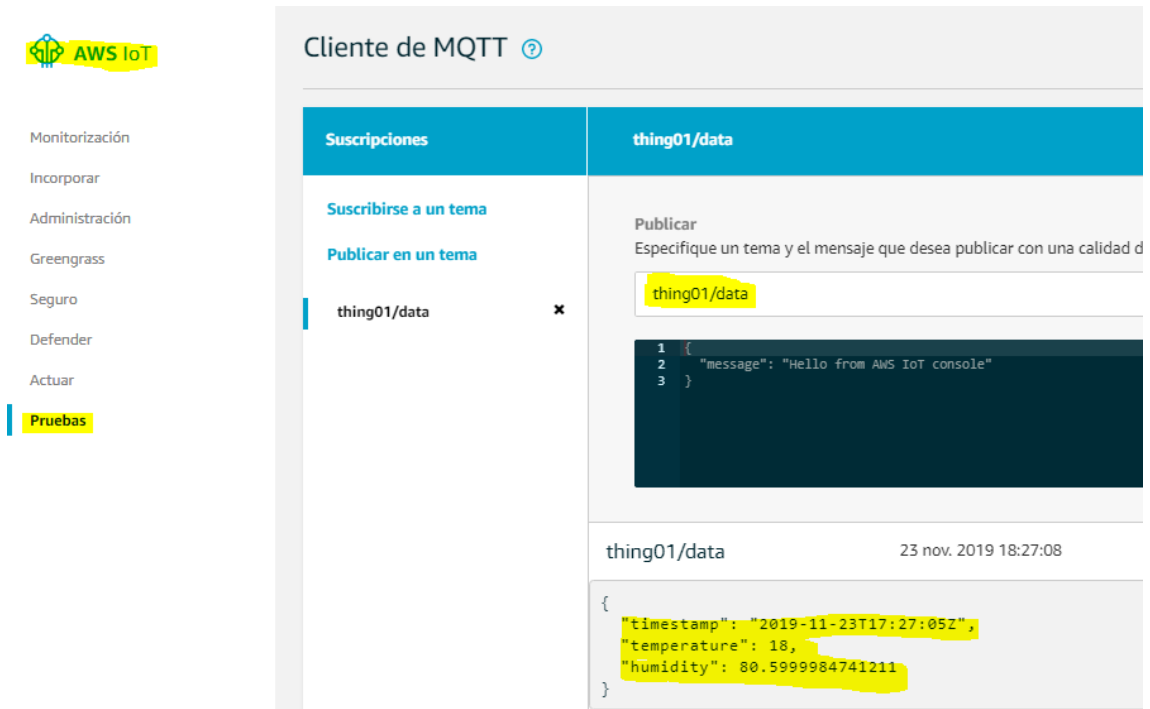


Figura 4.14. Prueba de funcionamiento de envío de datos a AWS IoT. Fuente: Pantallazo PC

Una vez realizado esto pasamos a ejecutar el script realdataPub.py en la RPi. Para saber que todo funciona correctamente debemos ver los datos sensados en la ventana de pruebas tal y como se muestra en la Figura 4.14.

4.3. Configuración de DynamoDB y creación de reglas en AWS IoT.

Una vez hayamos llegado a este paso ya estamos preparados para almacenar los datos capturados y almacenarlos en una base de datos no relacional (DynamoDB). Es posible almacenar los datos en cualquier base de datos, sea relacional (SQL) o no relacional (NoSQL) como ya se ha explicado en la parte teórica.

Para esta arquitectura se ha elegido DynamoDB, ya que forma parte de la nube de AWS y continuamos manteniendo todo el en la nube de Amazon. A continuación, se explica cómo crear una Regla en AWS IoT y configurar AWS DyanamODB, para almacenar los datos sensados por nuestra RPi.

Sin salir del servicio de AWS IoT, le damos clic donde pone actuar, tal y como se muestra en la Figura 4.14. Una vez en la opción actuar le damos a reglas. Si no tenemos ninguna regla creada se nos dará la opción de crear una regla.

- AWS IoT (IoT Core) >> Actuar >> Reglas >> Crear

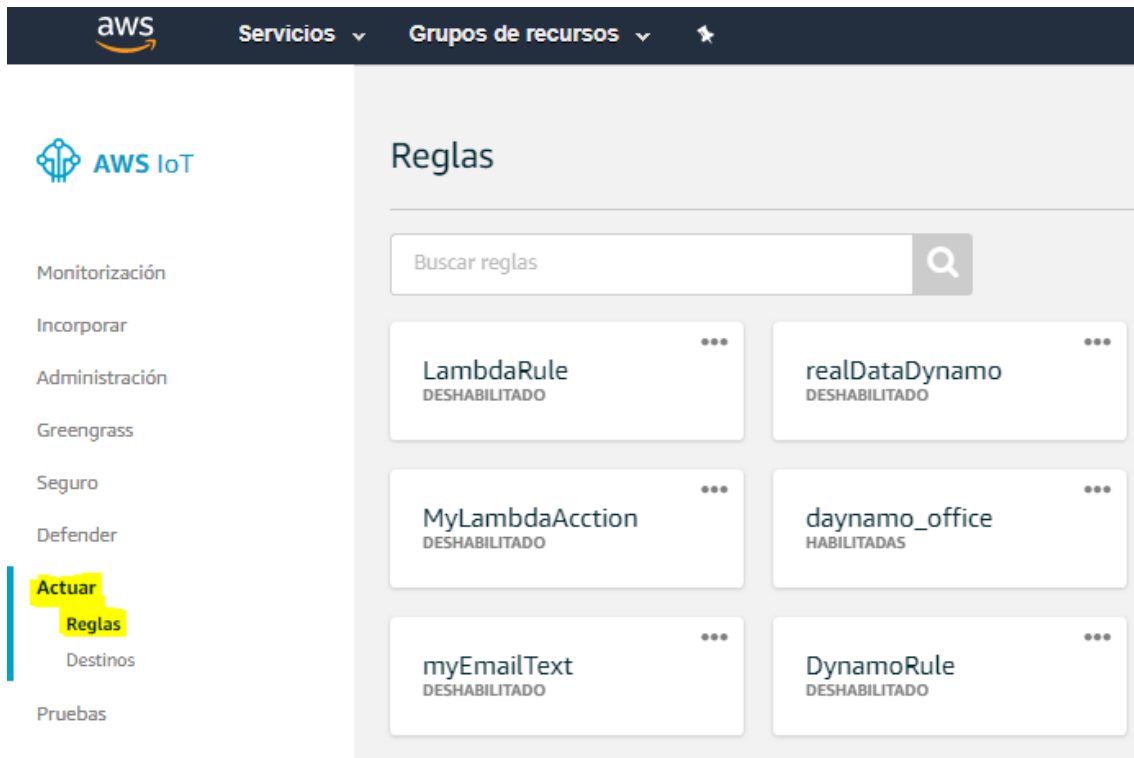


Figura 4.15. Creación de una Regla para el objeto IoT. Fuente: Pantallazo PC

Una regla es lo que nos permite enlazar el servicio de AWS IoT con otros servicios además de DynamoDB. Así además de almacenar los datos, podemos por ejemplo, enlazar el objeto creado en AWS IoT con el servicio de mensajería de AWS SNS, de tal manera que recibamos un mensaje en el correo electrónico o en el móvil al recibir los datos de la RPi.

Una vez le damos a crear una regla nos encontramos con la imagen de la Figura 4.16. En esta ventana, debemos hacer cuatros cosas muy simples:

1. Poner un nombre
2. Poner una pequeña descripción
3. En la parte de “Instrucción de consulta de regla”, debemos poner una sentencia SQL, que nos permita sintonizar con el canal (tópico), con el que se está enviando la información en la RPi. En este caso pondremos lo siguiente:

```
>>> SELECT * FROM 'thing01/data'
```

Lo que está entre comillas es el tópico/canal en el que se envía la información en nuestra RPi.

4. En la opción definir una o varias acciones, seleccionamos la acción que deseamos. De todas las opciones elegimos “Insertar un mensaje en una tabla de DynamoDB”, que es exactamente lo que deseamos hacer.

Crear una regla

Cree una regla para evaluar los mensajes enviados por sus objetos y especifique lo que se debe hacer cuando se reciba un mensaje (por ejemplo, escribir datos en una tabla de DynamoDB o invocar una función Lambda).

Nombre

Descripción

Instrucción de consulta de regla

Indique el origen de los mensajes que desea procesar con esta regla.

Uso de la versión de SQL

2016-03-23

Instrucción de consulta de regla

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. Por ejemplo: SELECT temperatura FROM 'iot/topic' WHERE temperatura > 50. Para obtener más información sobre cómo crear una instrucción SQL, consulte [Referencia de SQL de AWS IoT](#).


```
1 SELECT * FROM 'iot/topic'
```

Figura 4.16. Configuración de la Regla. Fuente: Pantallazo PC

Seleccionar una acción


Seleccione una acción.

☒




Insertar un mensaje en una tabla de DynamoDB
DYNAMODB

☐




Dividir mensajes en varias columnas de una tabla de base de datos (DynamoDBv2)
DYNAMODBv2

☐



Invocar una función de Lambda para pasar los datos del mensaje
LAMBDA

☐



Enviar un mensaje como una notificación push SNS
SNS

Figura 4.17. Elección de la acción que deseamos. Fuente: Pantallazo PC

Una vez elegido la primera opción de DynamoDB, debemos configurarla, por lo que le damos a configurar acción. Una vez de dentro de configurar acción, le damos a crear recurso, lo que significa crear una Tabla en DynamoDB.

Esto nos abrirá una nueva pestaña en nuestro navegador, en el cual se nos mostrará el servicio de AWS DaynomoDB tal y como se muestra en la Figura 4.19. Debemos crear una tabla en DynamoDB, que es donde pondremos los datos enviados por la RPi.

The screenshot shows the 'Configurar acción' (Configure action) page in the AWS IAM console. The main heading is 'Insertar un mensaje en una tabla de DynamoDB' (Insert a message into a DynamoDB table). Below this, a note states: 'La tabla debe contener claves hash y de rango.' (The table must contain hash and range keys). The form includes a dropdown for '*Nombre de la tabla' (Table name) with the placeholder 'Seleccione un recurso' (Select a resource) and a 'Crear un nuevo recurso' (Create new resource) button. Below this, there are two rows of input fields for key configuration. The first row is for the hash key: '*Clave de partición' (Partition key) with 'El campo necesario no existe' (Required field does not exist), '*Tipo de clave hash' (Hash key type) with 'El campo necesario no existe' (Required field does not exist), and '*Valor de clave hash' (Hash key value) with an empty text box. The second row is for the range key: 'Clave de rango' (Range key) with 'El campo opcional no existe' (Optional field does not exist), 'Tipo de clave de rango' (Range key type) with 'El campo opcional no existe' (Optional field does not exist), and 'Valor de clave de rango' (Range key value) with an empty text box. At the bottom, there is a section 'Escribir datos del mensaje en esta columna' (Write message data in this column) with an empty text box, and an 'Operación' (Operation) dropdown with a question mark icon.

Figura 4.18. Creación de la tabla en DynamoDB. Fuente: Pantallazo PC

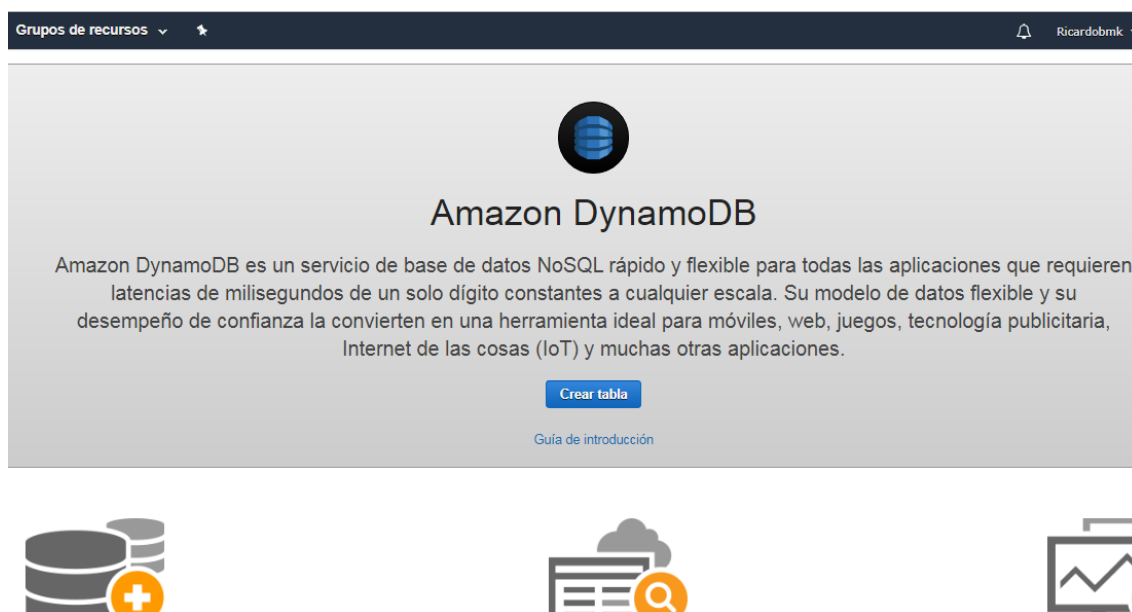


Figura 4.19. Página principal AWS DaynomoDB. Fuente: Pantallazo PC

5. Le damos a crear una nueva tabla. Una vez le hayamos dado a ese botón se nos aparecerán una imagen parecida a la que se muestra en la Figura 4.20, donde se son pedirá información como el nombre, clave de partición y clave de ordenación.

Crear una tabla de DynamoDB Tutorial ?

DynamoDB es una base de datos sin esquema que solo necesita un nombre de tabla y una clave principal. La clave principal de la tabla está compuesta de uno o dos atributos que identifican de manera inequívoca cada elemento, efectúan la partición de datos y ordenan los datos dentro de cada partición.

Nombre de la tabla*

Clave principal* Clave de partición

Cadena ?

☒ Añadir clave de ordenación

Cadena ?

Configuración de la tabla

La configuración predeterminada proporciona la forma más rápida de comenzar con la tabla. Puede modificar esta configuración predeterminada ahora o después de crear la tabla.

☒ Usar la configuración predeterminada

- No hay índices secundarios.
- Capacidad aprovisionada establecida en 5 lecturas y 5 escrituras.
- Alarmas básicas con umbral superior al 80% que usan el tema de SNS "dynamodb".
- Cifrado en reposo con el tipo de cifrado PREDETERMINADO.

No tiene la función necesaria para habilitar Auto Scaling de forma predeterminada.
Consulte Documentación.

+ Añadir etiquetas **NOVEDADES!**

Es posible que se apliquen cargos adicionales si superan las capas gratuitas de AWS para CloudWatch o Simple Notification Service. La configuración avanzada de la alarma está disponible en la consola de administración de CloudWatch.

Cancelar Crear

Figura 4.20. Creación de la tabla. Fuente: Pantallazo PC

La clave de partición y la clave de ordenación, se deben poner tal y como se muestra en la Figura 38, ya que de esta manera resulta más fácil acceder a la información que se encuentra dentro de la tabla.

DynamoDB es una base de datos sin esquema que solo necesita un nombre de tabla y una clave principal que identifican de manera inequívoca cada elemento, efectúan la partición de datos y ordenan los datos dentro de cada partición.

Nombre de la tabla* ?

Clave principal* Clave de partición

Cadena ?

☒ Añadir clave de ordenación

Número ?

Figura 4.21. Datos a incluir para crear la Tabla. Fuente: Pantallazo PC

6. Volvemos a la pestaña de AWS IoT y nos encontramos que ya podemos seleccionar la Tabla que hemos creado, y rellenar los siguientes datos que nos piden, tal y como se muestra en la Figura 4.22.

Configure action

Insert a message into a DynamoDB table

The table must contain Partition and Sort keys.

*Table name
 [Create a new resource](#)

*Partition key	*Partition key type	*Partition key value
<input type="text" value="id"/>	<input type="text" value="STRING"/>	<input type="text" value="\${topic()}"/>
Sort key	Sort key type	Sort key value
<input type="text" value="timestamp"/>	<input type="text" value="NUMBER"/>	<input type="text" value="\${timestamp()}"/>

Write message data to this column

Operation [?](#)

Figura 4.22. Configuración de la acción DynamoDB. Fuente: Pantallazo PC

Si nos fijamos en la misma Figura 4.22, vemos que hemos puesto **id** en **Partition key**, **STRING** en **Partition key type** y **\${topic()}** en **Partition key value**. Estos datos que hemos introducido en esta primera fila significa que el tópic con el que se envíen los datos se establece como la clave principal. Esto nos facilitará las cosas a la hora de realizar consultas con JavaScript.

En la siguiente fila de la misma Figura 4.22, hemos puesto **timestamp** en **Sort key**, **NUMBER** en **Sort key type** y **\${timestamp()}** en **Sort key value**. Estos datos que hemos introducido en esta segunda fila nos permite poner el contenido de timestamp (dato de la fecha) en este campo.

Por último, tenemos la tercera fila, en esta fila pondremos el nombre que queramos. Dentro de ese mensaje se encontrarán todos los datos que envíe la RPi.

Le damos a añadir acción y con esto habrías acabado de configurar la acción.

El resultado de configurar la regla debe ser el que se muestra en la Figura 4.23. Cuando le damos clic a la regla que hemos creado debe aparecernos algo como lo que se muestra en esa imagen.

REGLA

dashboard_dynamo_iot

HABILIDADES

Acciones ▾

Información general

Descripción

Editar

Tags

trying to make a dashboar with dynamo db and IoT.

Instrucción de consulta de regla

Editar


El origen de los mensajes que desea procesar con esta regla.

```
SELECT * FROM 'thing01/data'
```

Uso de la versión de SQL 2016-03-23

Acciones

Las acciones son lo que ocurre cuando se activa una regla. [Más información](#)



Insertar un mensaje en una tabla de DynamoDB

dashboard_table

Eliminar

Editar ▸

Añadir acción

Acción de error

Si lo desea, defina la acción que se ejecutará cuando se produzca un error al procesar la regla.

Añadir acción

Figura 4.23. Resultado final de crear la Regla. Fuente: Pantallazo PC

Ahora sólo nos falta verificar que efectivamente, al enviar los datos desde la RPi, realmente se almacenan los datos en la base de datos DynamoDB. Para realizar esta prueba bastará con ejecutar nuevamente el script de Python (realdatapub14.py), que ya se ha explicado en las páginas anteriores. Una vez hayamos enviado 3 o 4 datos, detenemos el script y vamos a mirar la tabla creada en DynamoDB, para ver si contiene algún dato.

El resultado debe ser parecido al que se muestra en la Figura 4.24.

aws

Services ▾

Resource Groups ▾

DynamoDB

Dashboard

Tables

Backups

Reserved capacity

Preferences

DAX

Dashboard

Clusters

Subnet groups

Parameter groups

Events

Create table

Delete table

Filter by table name

Choose a table ...

Actions

Name

dashboard_table

dashboard_table

Close

Overview

Items

Metrics

Alarms

Capacity

Indexes

Global Tables

Backups

Contributor Insights

Triggers

Access o

Create item

Actions ▾

Scan: [Table] dashboard_table: id, timestamp

Scan

[Table] dashboard_table: id, timestamp

Add filter

Start search

id	timestamp	payload
thing01/data	1575712100586	{"humidity": {"N": "63"}, "temperature": {"N": "20.100000381469727"}, ...
thing01/data	1575712107648	{"humidity": {"N": "63.099998474121094"}, "temperature": {"N": "20.10...
thing01/data	1575712114705	{"humidity": {"N": "63.099998474121094"}, "temperature": {"N": "20.10...
thing01/data	1575712219876	{"humidity": {"N": "61.900001525878906"}, "temperature": {"N": "20.10...

Figura 4.24. Datos de la tabla creada. Fuente: Pantallazo PC

88

Cada fila, se conoce como un Item y cada Item contiene los datos enviados. Para entenderlo mejor vamos a hacer clic en alguna de las filas, para ver el contenido de una de ellas.

Ya sabemos que los datos se envían en formato JSON. DynamoDB, muestra los datos de una manera mucho más fácil de entender, pero siguen estando en formato JSON. Como podemos ver en la figura 4.25, tenemos el **id**, **payload** que contiene la humedad, la temperatura y el **timestamp** en formato String y por último tenemos nuevamente el **timestamp** en formato numérico.

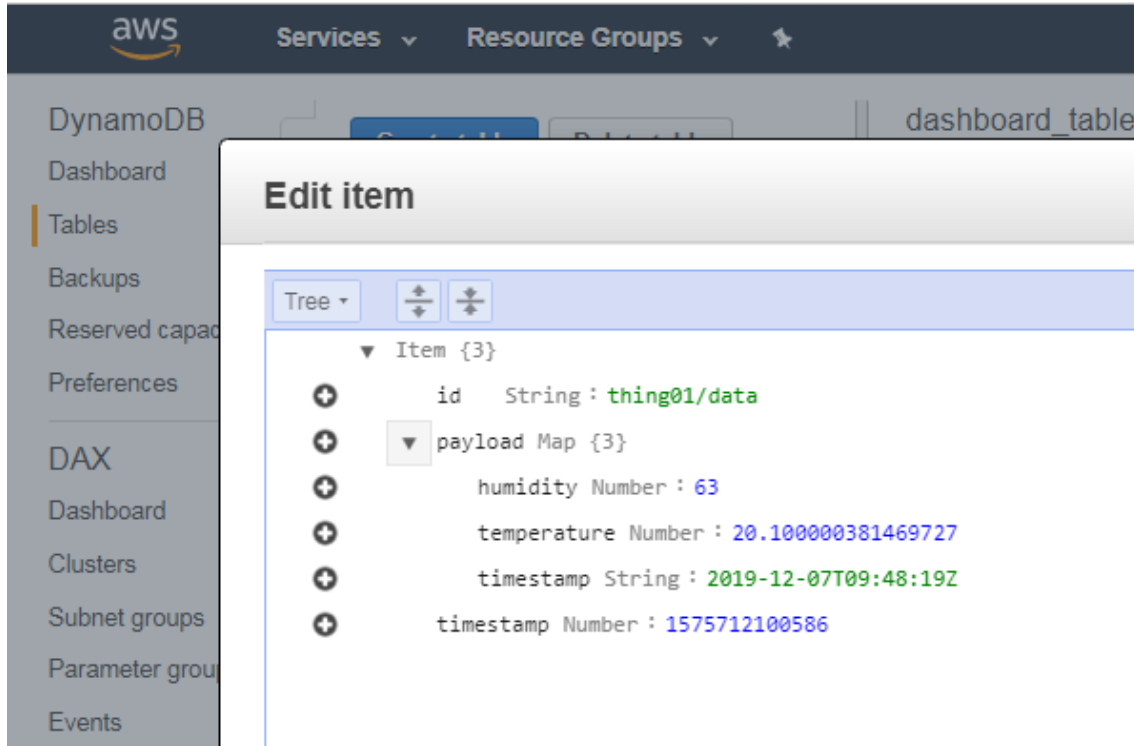


Figura 4.25. Dentro de un Item o paquete de información. Fuente: Pantallazo PC

Ya tenemos esta parte realizada y funcionando correctamente. En siguientes apartados vamos a crear la web que muestra gráficamente estos datos y lo asociaremos con un dominio real para su visualización en la red de redes.

4.4. Creación de la plataforma Web y configuración de S3.

La página web que se ha creado, se ha realizado con HTML, JavaScript, ChartJS, CSS y Bootstrap 4.

El código HTML, CSS y Bootstrap 4, no tiene mucho sentido explicarlo ya que es un tema más relacionado con la presentación del contenido y no aportaría mucho valor. Basta con poner en internet como hacer un web con estos lenguajes y tendremos información sobre cómo hacerlo. Por lo que no explicaremos esta parte, preferimos mostrar todo el código a través de Github: ([Link de acceso a mi github](#)).

Hay dos partes importantes en el código en HTML, que es la parte que permite relacionar acceder al SDK de JavaScript, sin esa línea de código la página web no puede acceder al kit de

desarrollo creado por AWS para utilizar AWS desde la web. La otra línea de código corresponde con la librería de ChartJs, que permite realizar los gráficos.

```
<!-- AWS SDK for JS CDN -->

<script src="https://sdk.amazonaws.com/js/aws-sdk-2.7.16.min.js"></script>

<!-- ChartJS 2.8 CDN -->

<script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>
```

Script 4.3. Acceso a SDK JavaScript y librería ChartJS.

La parte que, sí merece la pena explicar y está relacionado con el código y la programación, es la parte relacionada con JavaScript y ChartJS.

Como ya se ha explicado anteriormente, se trata de una aplicación sin servidor, por lo que no hay una manera habitual de realizar consultas a una base de datos tal y como lo haríamos con Python o Java. La forma de acceder a la base de datos a través de la plataforma se realiza únicamente con código JavaScript y es lo que pasaremos a explicar a continuación.

Credenciales de acceso a DynamoDB:

```
AWS.config.update({
    region: "us-west-2",
    // The endpoint should point to the local or remote computer where DynamoDB
    (downloadable) is running.
    endpoint: 'https://dynamodb.us-east-1.amazonaws.com',
    /*
        accessKeyId and secretAccessKey defaults can be used while using the downloadable
        version of DynamoDB.
        For security reasons, do not store AWS Credentials in your files. Use Amazon
        Cognito instead.
    */
    accessKeyId: "fakeMyKeyId",
    secretAccessKey: "fakeSecretAccessKey"
});
```

Script 4.4. Credenciales de acceso a DynamoDB.

Para poder acceder a la base de datos de DynamoDB se puede hacer de muchas formas, nosotros hemos optado por modificar la opción menos recomendada para hacerla un poco más recomendada.

AWS IAM, permite dar acceso a terceros, para que utilicen tu cuenta de AWS. Esto puede ser útil a la hora de trabajar en equipo o cuando tienes un problema con tu cuenta y te gustaría que un profesional te lo solucione o algo por el estilo.

AWS IAM, nos permite crear un usuario y dar un acceso total a toda nuestra cuenta (no nos interesa esta opción) o dar un acceso limitado según nuestras necesidades (la opción que nos interesa).

Por lo tanto, vamos a crear un usuario que únicamente pueda leer las tablas creadas en DynamoDB. Para ello realizaremos los siguientes pasos:

1. Vamos al servicio de AWS IAM, le damos a “Users” y después a “Add user”. En la Figura 4.26, se puede ver lo fácil que resulta acceder a esta opción.

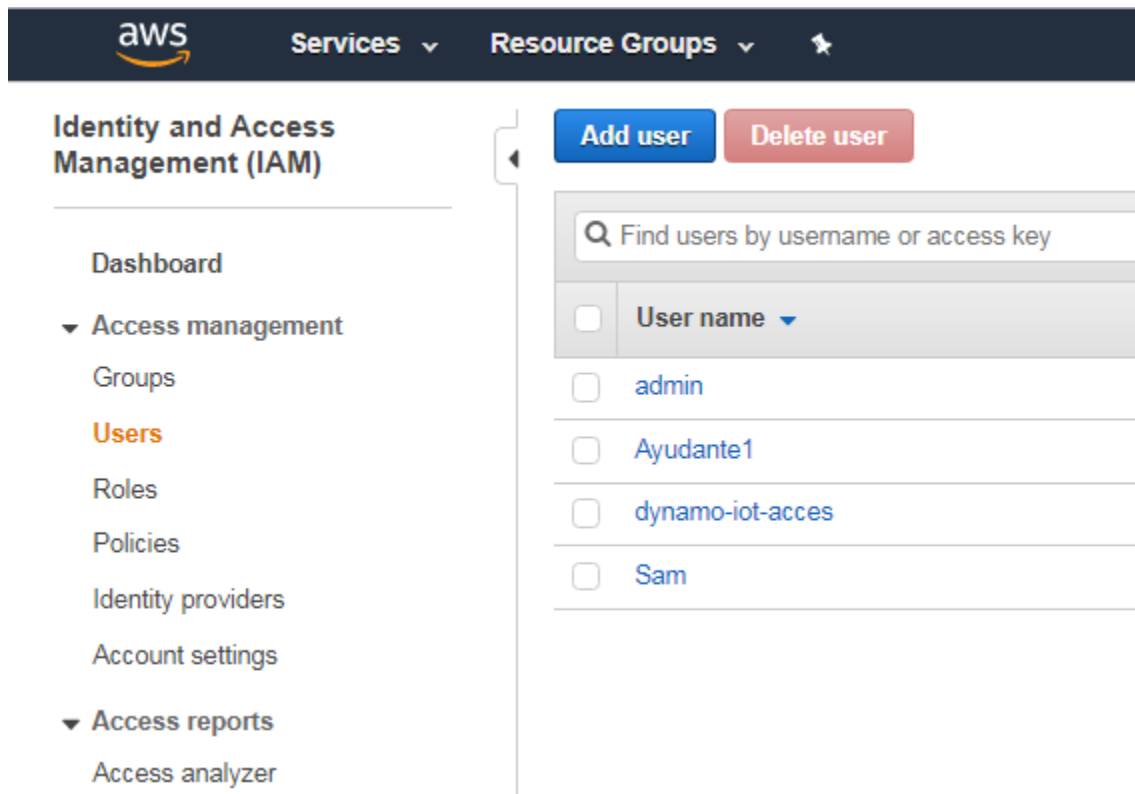


Figura 4.26. Creación del usuario con AWS IAM. Fuente: Pantallazo PC

2. Al acceder a la opción de añadir usuario, se nos pide un nombre y al modo de acceso el cual queremos que accedan los usuarios de nuestra cuenta. Le damos clic a “Programatic access”, esta opción nos permite entrar a través de una SDK (Software Development Kid), que corresponde con lo que utilizaremos, el SDK de JavaScript proveído por AWS. En la figura 4.27, tenemos una imagen, para hacernos una idea.

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* dynamoDB

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Access type* ☒ **Programmatic access**
 Enables an **access key ID** and **secret access key** for the AWS API, CLI, **SDK**, and other development tools.
- ☐ **AWS Management Console access**
 Enables a **password** that allows users to sign-in to the AWS Management Console.

Figura 4.27. Configuración 1 del usuario. Nombre y tipo de acceso. Fuente: Pantallazo PC

3. Una vez hemos realizado este paso se nos pide la política del usuario, como ya sabemos la política que se asocia a un usuario da permiso o limita lo que puede llegar a realizar un usuario con nuestra cuenta. Le damos a crear una política (Figura 4.28), si no queremos complicarnos elegimos una política que ya existe y que se amolde a nuestras necesidades (Figura 4.29).

Add user

1 2 3 4 5

▼ Set permissions

Add user to group

Copy permissions from existing user

Attach existing policies directly

Create policy

Filter policies ▼

Q Search

Showing 543 results

	Policy name ▼	Type	Used as
<input type="checkbox"/>	AdministratorAccess	Job function	Permissions policy (3)
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessReadOnlyAccess	AWS managed	None
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	AWS managed	None
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	AWS managed	None

► Set permissions boundary

[Cancel](#)

[Previous](#)

[Next: Tags](#)

Figura 4.28. Configuración 2 del usuario. Creación de la política de acceso 1. Fuente: Pantallazo PC

En caso de que se opte por una política ya existente, basta con Filtrar en el buscador poniendo el nombre DynamoDB y nos encontramos con una política con el nombre **AmazonDynamoDBReadOnlyAccess**, que es el que dará al usuario únicamente la posibilidad de leer la base de datos DynamoDB.

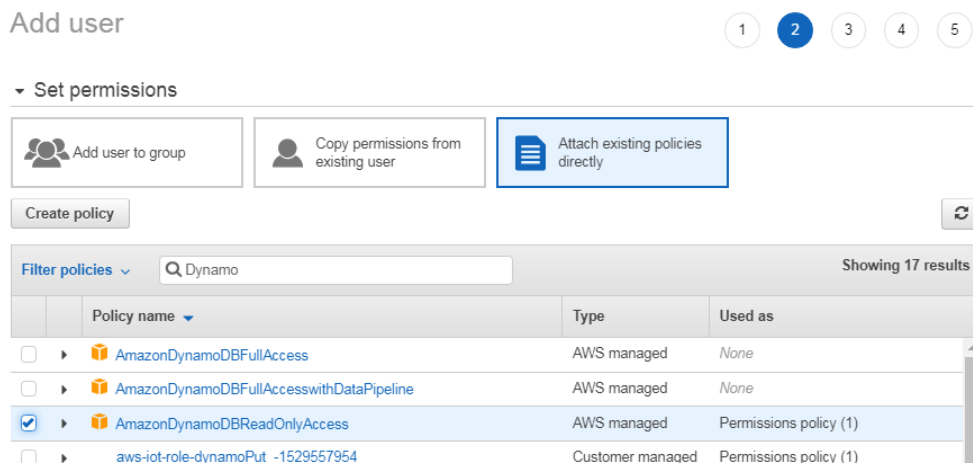


Figura 4.29. Configuración 2 del usuario. Elección de una política ya existente. Fuente: Pantallazo PC

En la Figura 4.30, podemos ver los pasos que se siguen para crear una política en caso que lo deseemos hacer de esta manera, AWS es muy intuitivo y nos da la opción de configurar esta opción a base de clics para aquellos que no quieran complicarse configurarlo picando código en JSON. Basta con poner en la opción "Action", **Read** y dejar todo lo demás tal y como se muestra en la imagen.

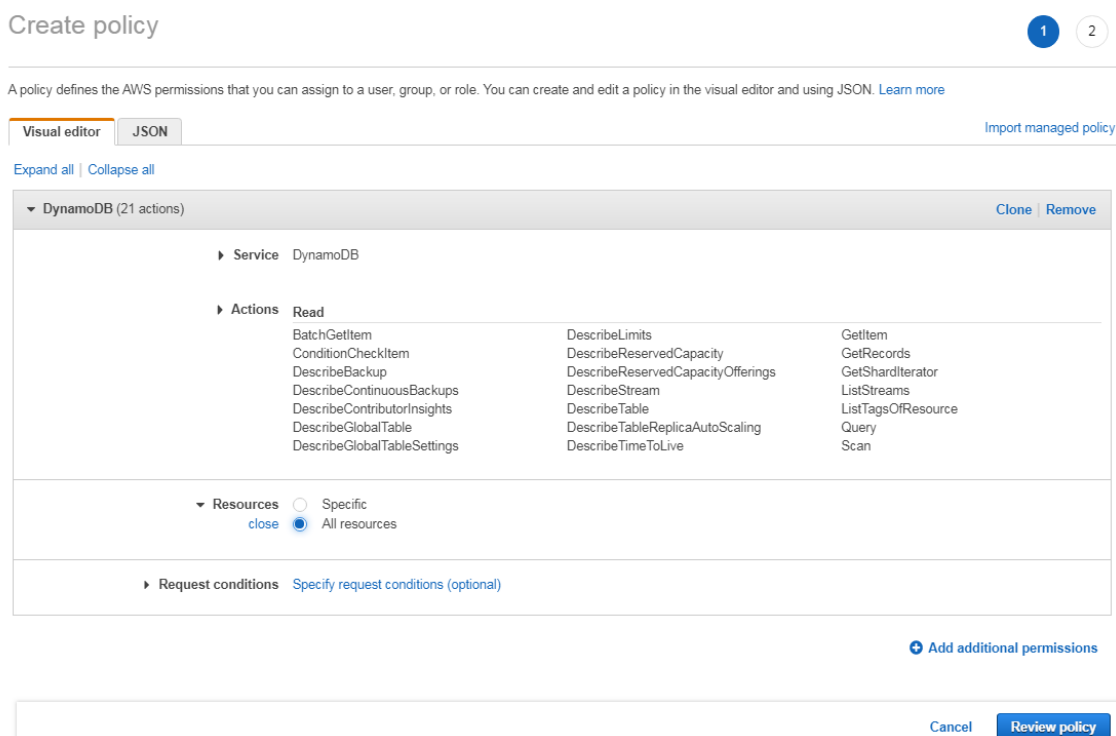


Figura 4.30. Configuración 2 del usuario. Creación de la política de acceso 2. Fuente: Pantallazo PC

Create policy

1 2

Review policy

Name*

Use alphanumeric and '+-._@-_' characters. Maximum 128 characters.

Description

Maximum 1000 characters. Use alphanumeric and '+-._@-_' characters.

Summary

Filter			
Service	Access level	Resource	Request condition
Allow (1 of 214 services) Show remaining 213			
DynamoDB	Full: Read	All resources	None

* Required

Cancel

Previous

Create policy

Figura 4.31. Configuración 2 del usuario. Creación de la política de acceso 3. Fuente: Pantallazo PC

Continuando con la creación manual de nuestra política, en la siguiente imagen (Figura 4.31). Se nos pide el nombre que queremos darle a la política y una pequeña descripción si lo deseamos. Con esto ya habríamos realizado todos los pasos para crear nuestra propia política y con nuestros propios gustos. Ya podemos asociarla al usuarios creado buscando la política igual que se ha hecho con la política por defecto.

Tanto la política creada por la opción por defecto como la configurada manualmente funcionan según lo deseado.

4. Una vez creado el usuario con la única posibilidad de leer la información que disponemos en nuestra base de datos ya podemos descargarnos las credenciales del usuario tal y como podemos ver en la imagen (Figura 4.32).

Estas credenciales son las que se ponen en el código de JavaScript para acceder con el SDK de JavaScript.

Add user

1 2 3 4 5

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://ricardobmk.signin.aws.amazon.com/console>

Download .csv

User	Access key ID	Secret access key
▶ DynamoDB_ReadOnly	AKIAZ3JOYZXEDLXSXTFZ	***** Show

Figura 4.32. Configuración 2 del usuario. Descargar credenciales de acceso. Fuente: Pantallazo PC

Como podemos ver, esto no es muy seguro y no se recomienda, pero es útil para el propósito de prueba de concepto.

Ahora que ya hemos configurado la política, vamos a pasar a explicar las siguientes partes del código en JavaScript.

```
var params = {
  TableName: "dashboard_table",
  KeyConditionExpression: "#id = :iotTopic AND #ts >= :datum",
  ExpressionAttributeNames: {
    "#id": "id",
    "#ts": "timestamp"
  },
  ExpressionAttributeValues: {
    ":iotTopic": "thing01/data",
    ":datum": datumVal
  }
};
```

Script 4.5. Algoritmo de búsqueda en DynamoDB.

El Script 4.5, lo que nos permite es realizar un “query” o consulta en la base de datos no relacional de DynamoDB. Esto consiste en poner las instrucciones necesarias para buscar la información que deseamos. Si nos fijamos atentamente nos encontraremos con lo siguiente:

- **TableName**, nos permite buscar por el nombre de nuestra tabla creada en la base de datos.
- **KeyConditionExpression**, nos permite decir que queremos buscar por “id” y por “ts”. Decimos que el “id” corresponda con el tópico que hemos creado y “ts” corresponda con timestamp.
- **ExpressionAttributeValues**, le damos los valores a las variables que hemos creado anteriormente. **iotTopic**, le damos el nombre del tópico en el que estamos enviando la información por MQTT. **Datum**, le damos la línea temporal en el que queremos ver los datos.

DatumVal es una variable que ya se ha creado anteriormente y da información de un periodo de tiempo.

En otras palabras, lo que hace este código es decir entra en la tabla `dashboard_table` busca el tópico `thing01/data` en todos los periodos de tiempo que comprendan `datumVal`. De esta manera es posible realizar consultas a la base de datos que puedan ser de los últimos dos, tres o mes si lo deseamos.

Configurar el gráfico con ChartJS

```
var tctx = $("#temperaturegraph").get(0).getContext("2d");
var hctx = $("#humiditygraph").get(0).getContext("2d");

var options = {
  responsive: true,
  showLines: true,
  legend: {
    display: false
  },
  scales: {
    xAxes: [{
      display: true
    }],
    yAxes: [{
      ticks: {
        beginAtZero: true
      }
    }]
  }
};
```

Script 4.6. Código para crear los gráficos con ChartJS.

Las línea del Script 4.6, lo que hacen es crear dos variables (tctx, hctx) que se utilizan más adelante con una características adecuadas para realizar los gráficos. Una para la temperatura y otra para la humedad. Las siguientes líneas de código lo que hacen es configurar las características del gráfico, si nos fijamos que simplemente definimos con true o false aquellas características que queremos que se muestren en el gráfico.

```
$(function() {  
    getData();  
    //$.ajaxSetup({ cache: false });  
    setInterval(getData, 500);  
});
```

Script 4.7. función que actualiza la función getData().

El Script 4.9, es una función que se dedica a ejecutar la función `getData()`, cada 500 milisegundos. La función `getData()` es la función que recoge y almacena los datos en variables a medida que se van creando nuevos datos.

```
var tinit = {  
    labels: [],  
    datasets: [  
        {  
            label: "Temperature °C",  
            backgroundColor: 'rgba(204,229,255,0.5)',  
            borderColor: "#3e95cd",  
            data: []  
        }  
    ]  
};
```

Script 4.8. Configuración de las características del gráfico.

Las linea de Script 4.8, nos permiten continuar configurando el gráfico, pero no han un nivel visual sino que nos permiten poner los datos que se encuentran en la base de datos a modo de gráficos ponerles una etiqueta y un color de fondo.

```
function getData() {  
    docClient.query(params, function(err, data) {  
        if (err) {  
            console.log(err);  
            return null;  
        } else {  
            var temperatureValues = [];  
            var humidityValues = [];  
            //var timeValues = [];  
            var labelValues = [];  
  
            var temperatureRead = 0.0;  
            var humidityRead = 0.0;  
            //timeRead = "";  
  
            for (var i in data['Items']) {  
                temperatureRead = parseFloat(data['Items'][i]['payload']['temperature']);  
                humidityRead = parseFloat(data['Items'][i]['payload']['humidity']);  
                //timeRead = new Date(data['Items'][i]['timestamp']);  
                timeRead = data['Items'][i]['payload']['timestamp'];  
  
                temperatureValues.push(temperatureRead);  
                humidityValues.push(humidityRead);  
                labelValues.push(timeRead);  
            }  
        }  
        temperaturegraph.data.labels = labelValues;  
        temperaturegraph.data.datasets[0].data = temperatureValues;  
        humiditygraph.data.labels = labelValues;  
        humiditygraph.data.datasets[0].data = humidityValues;  
  
        temperaturegraph.update();  
        humiditygraph.update();  
    });  
}
```

Script 4.9. Función getData().

El Script 4.9, es sobre la función `getData()`, probablemente la más importante de todo el proyecto y la que más dificultad se ha tenido en dar con la manera de hacerlo, por que relaciona cosas de la librería ChartJS, JavaScript y DynamoDB. Lo más importante de la función es el bucle `for` y la manera en que se accede al dato en concreto que estamos buscando.

Para poder entender la función es necesario tener muy claro los fundamentos de JavaScript. Lo que hace esta función es simplemente almacenar en objetos correspondientes a la variable `sensada` todos los datos que se van capturando por el sensor/RPi.

Se muestran únicamente partes importantes de código. Para acceder al código completo ir a la siguiente ruta den Github donde se encuentra todo el código:

- TFG_IoTProject >> js >> chartOne.js

Configuración de AWS S3.

Una vez tenemos una pequeña idea del funcionamiento del código y como están estructuradas las carpetas es hora de utilizar el servicio AWS S3, que lo que haremos simplemente es crear dos cubos (buckets) con los nombres adecuados y subir todos estos archivos en uno de los cubos. Recordar que AWS S3, es simplemente una unidad de almacenamiento como lo pueden ser Google Drive o Dropbox, añadiendo unas cuantas funcionalidades más. A continuación, se explica paso a paso cómo hacer esto.

El nombre del bucket que creemos debe ser igual al nombre del dominio que hemos comprado.

1. Vamos al servicio y le damos a “Create bucket”
2. Ponemos el nombre y le damos directamente a “Create” dejando todo por defecto. Ya configuraremos lo necesario al entrar en la configuración de ese “Bucket” en concreto.

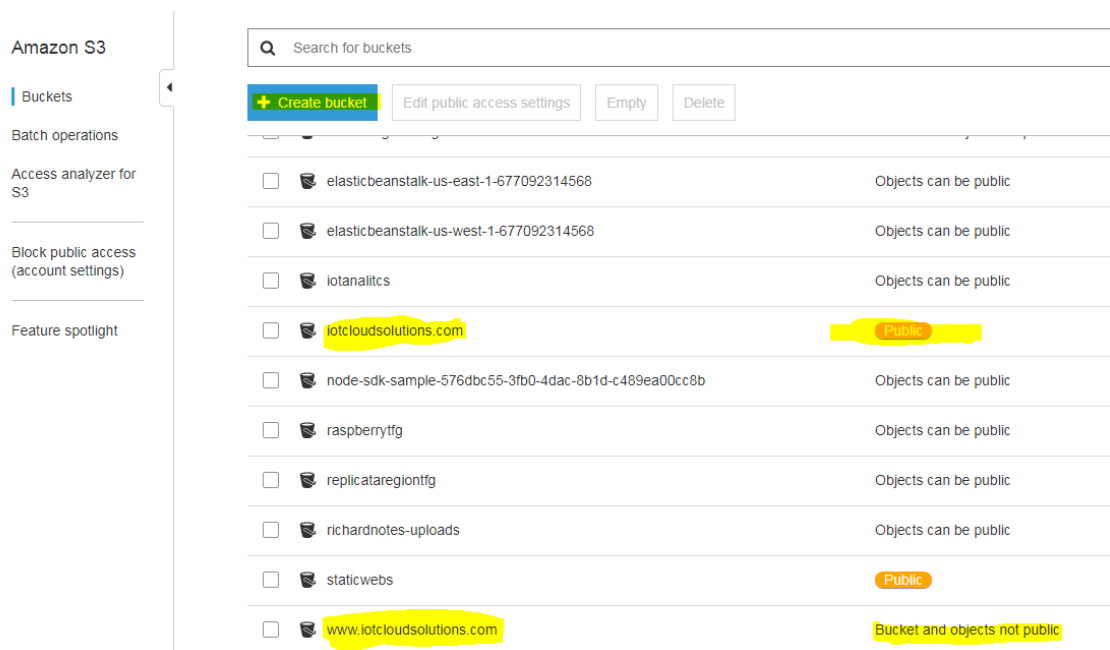


Figura 4.33. Creación del Bucket. Fuente: Pantallazo PC

3. Hacemos lo mismo nuevamente, pero esta vez creamos un “bucket”, con `www.[nombreElegido].com`. Esto es necesario para crear un redireccionameinto de nuestra página web en caso que alguien decida entrar poniendo `www`, cosa que rara vez ocurre.

Figura 4.34. Nombre del Bucket. Fuente: Pantallazo PC

<input type="checkbox"/>	Name ▾	Last modified ▾
<input type="checkbox"/>	.idea	--
<input type="checkbox"/>	css	--
<input type="checkbox"/>	imgs	--
<input type="checkbox"/>	js	--
<input type="checkbox"/>	sass	--
<input type="checkbox"/>	Charts.html	Dec 7, 2019 1:31:55 PM GMT+0100
<input type="checkbox"/>	index.html	Dec 7, 2019 1:31:55 PM GMT+0100

Figura 4.35. Archivos dentro del Bucket. Fuente: Pantallazo PC

Una vez que hemos creado los dos “Buckets”, vamos al bucket principal (iotcloudsolutions.com) y cargamos todos los archivos relacionados con la página web tal y como se muestra en la Figura 4.35. Con esto tendríamos los archivos dentro de AWS S3 en el bucket iotcloudsolutions.com, lo que sería una unidad de almacenamiento cualquier en la nube como drive o Dropbox.

Es necesario configurarlo un poco más para que nos permita utilizarlos como si fuera un hospedaje web. Para ello debemos ir a Propiedades y hacer clic en la opción “Static Web hosting” tal y como se muestra en la figura 4.36.

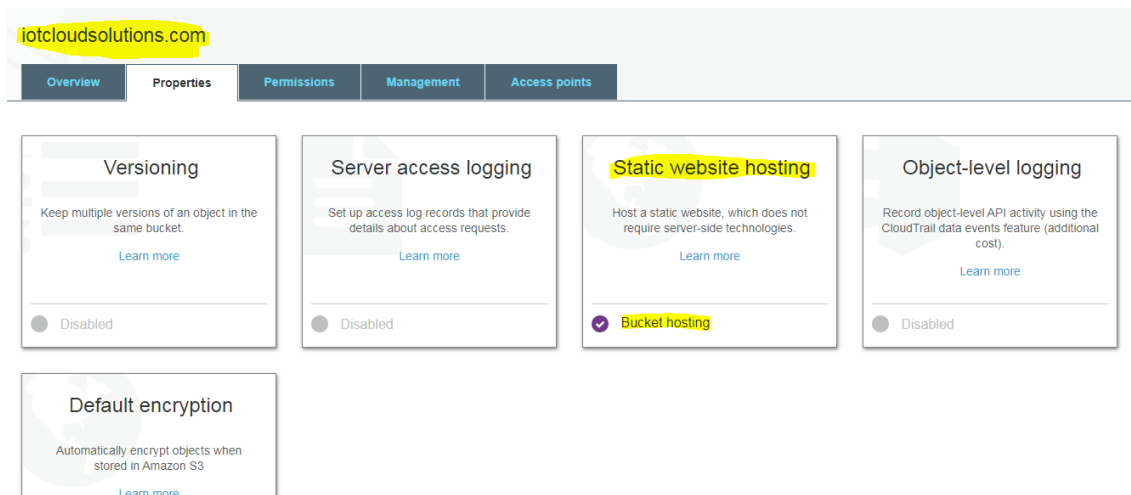


Figura 4.36. Configuración 1 del hospedaje de la Web. Fuente: Pantallazo PC

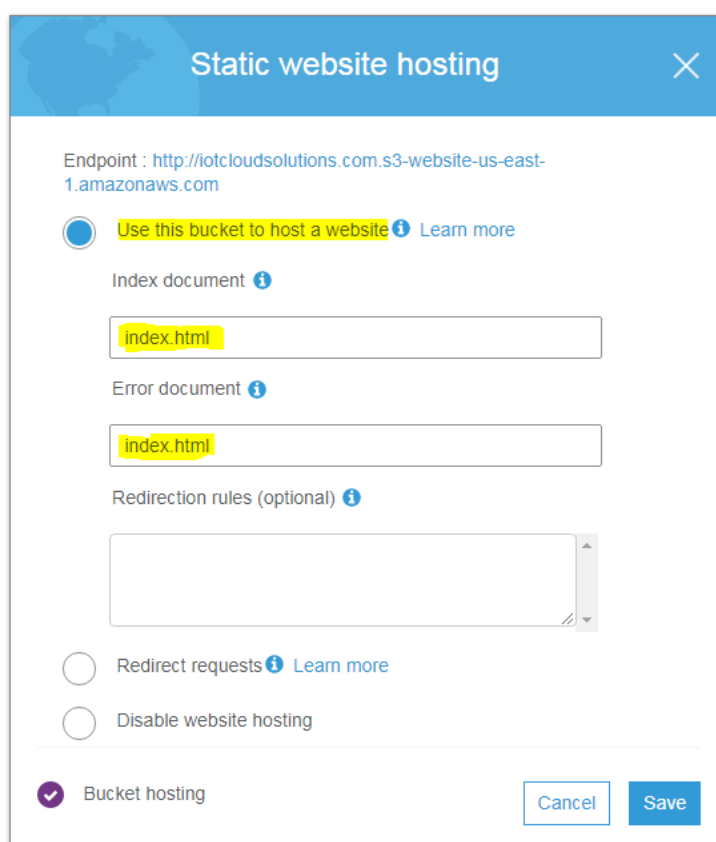


Figura 4.37. Configuración 2 del hospedaje de la Web. Fuente: Pantallazo PC

En la Figura 4.37, se nos muestra lo que nos encontramos al entrar en la opción del hospedaje web. Basta con configurarlo tal y como se muestra en la imagen. Lo que se está haciendo, es relacionar un link creado automáticamente por AWS S3, con el archivo principal de nuestro bucket (index.html).

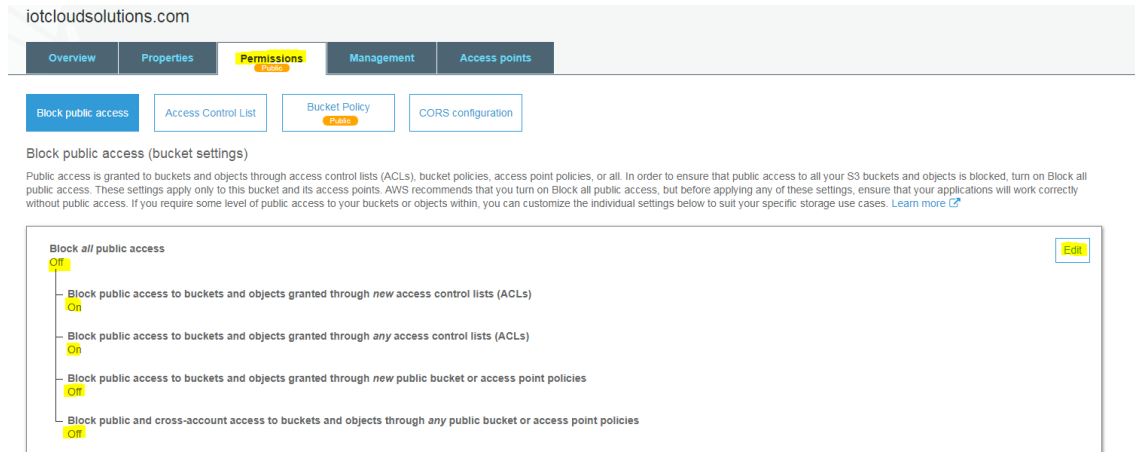


Figura 4.38. Configuración 1 del acceso a la Web. Fuente: Pantallazo PC

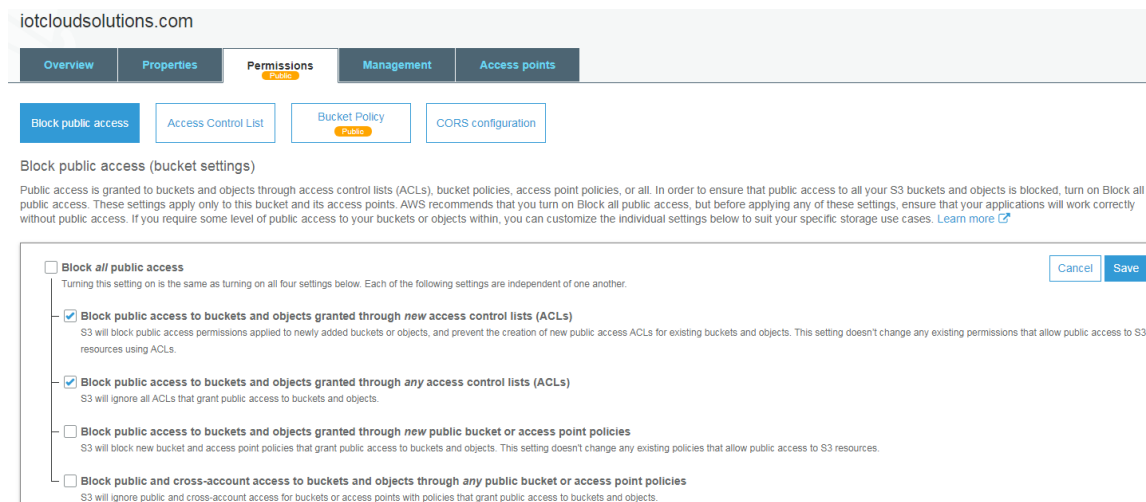


Figura 4.39. Configuración 2 del acceso a la Web. Fuente: Pantallazo PC

Una vez hemos configurado esta parte vamos a Permissions y lo configuramos para que sea público. Al tratarse de una página web necesitamos que cualquier persona pueda acceder a ella si dispone del link de acceso. Configurar esto es muy fácil y basta con hacerlo tal y como se muestra en la Figura 4.40.

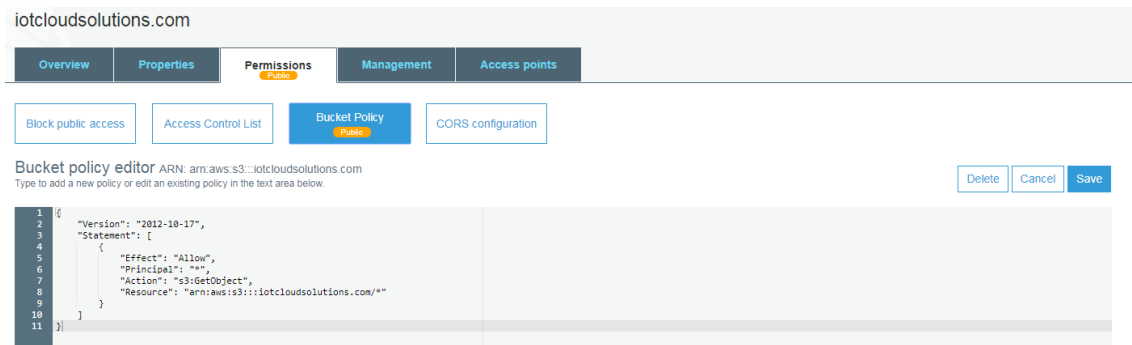


Figura 4.40. Configuración de la política de hospedaje

En la misma pestaña de “Permissions” vamos a “Bucket policy”, o ponemos el código JSON, que se muestra en el código X. Esto lo que hace es configurar una política de acceso total, que dé permiso a utilizar el bucket creado en AWS S3 con el nombre **iotcloudsolutions.com**

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::iotcloudsolutions.com/*"
    }
  ]
}

```

Script 4.10. Configuración 1 del acceso al bucket.

Para probar si está configurado correctamente, basta con probar el link que nos proporciona AWS S3 para acceder a nuestra web. Figura 4.41, al darle clic al link nos debe llevar a la página web que hemos creado, concretamente al archivo index.html, visto desde el navegador web.

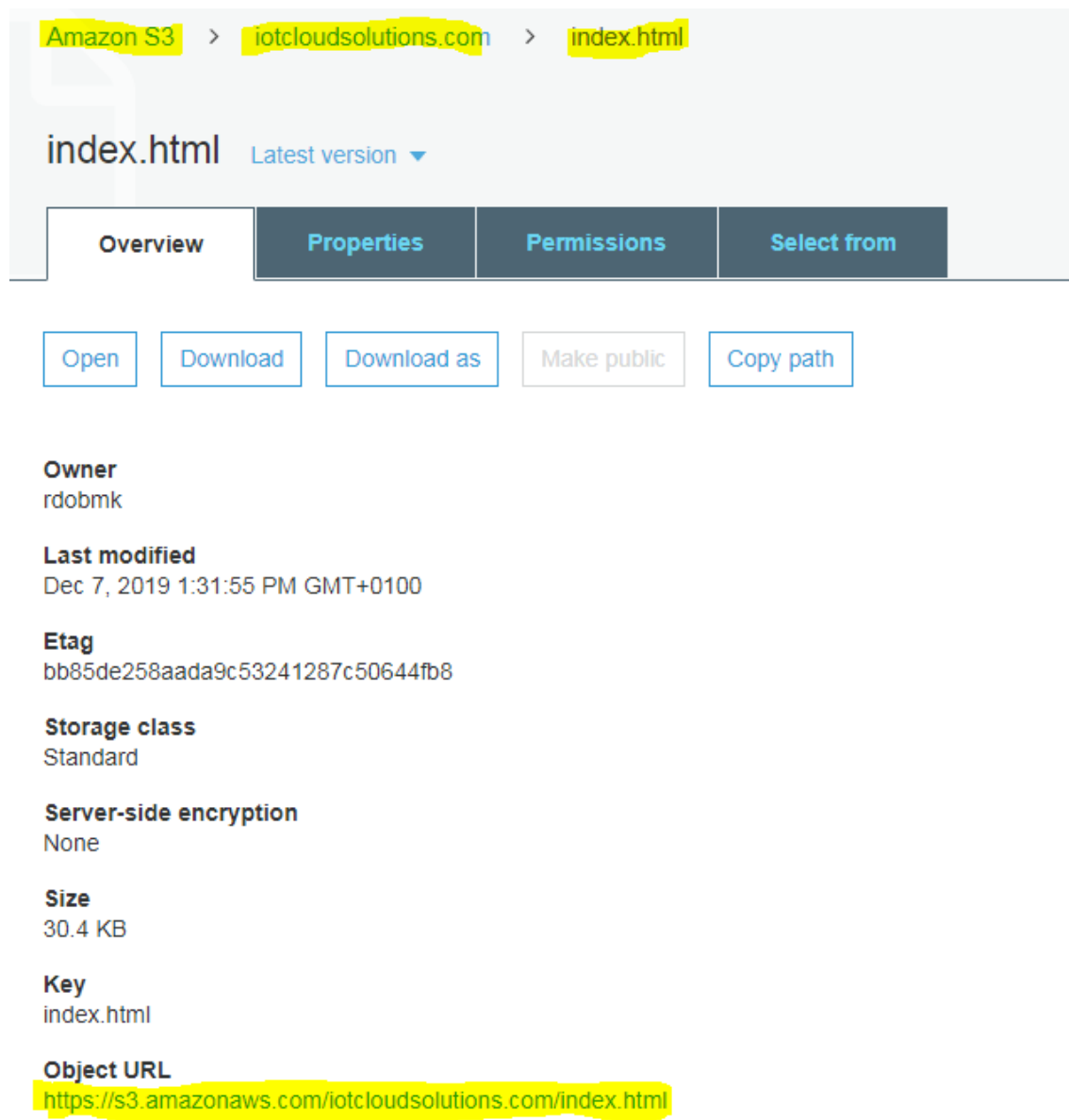


Figura 4.41. Link de AWS S3 para acceder a la Web. Fuente: Pantallazo PC

Si todo ha funcionado correctamente, toca configurar el segundo bucket, con el nombre `www.iotcloudsolutions.com`. Daremos pasos similares a los anteriores, solo que esta vez en vez de seleccionar la opción “Use this bucket to host a website”, seleccionaremos la opción “redirect request” y pondremos el nombre del bucket que nos interesa redireccionar que en este caso es `iotcloudsolutions.com`. En las siguientes, Figuras se muestran imágenes que ayudan a visualizar esta explicación.

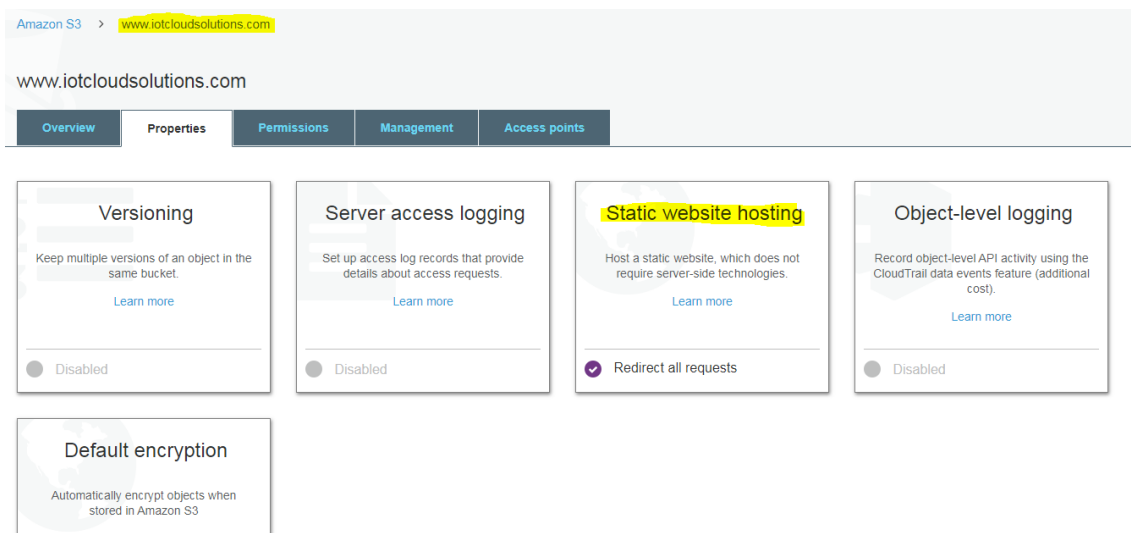


Figura 4.41. Configuración 1 de re-direccionamiento de la Web. Fuente: Pantallazo PC

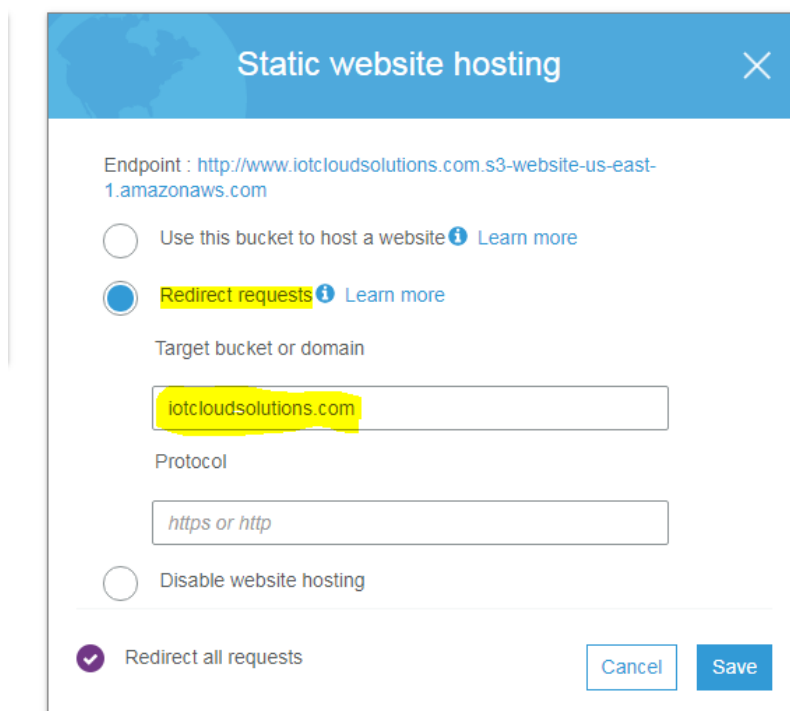


Figura 4.42. Configuración 2 de re-direccionamiento de la Web. Fuente: Pantallazo PC

Con esto estaríamos preparados para dar los siguientes pasos. Configuración de AWS Route53, para vincular el bucket a un link más fácil de recordar.

4.5. Configuración del Dominio con Route53.

En este apartado explicamos cómo le hemos dado un Dominio a la página web y lo hemos enlazado con S3.

Accedemos al servicio AWS Route53 y en el mismo Dashboard probamos si el dominio que deseamos está disponible para su uso. Figura 4.43.

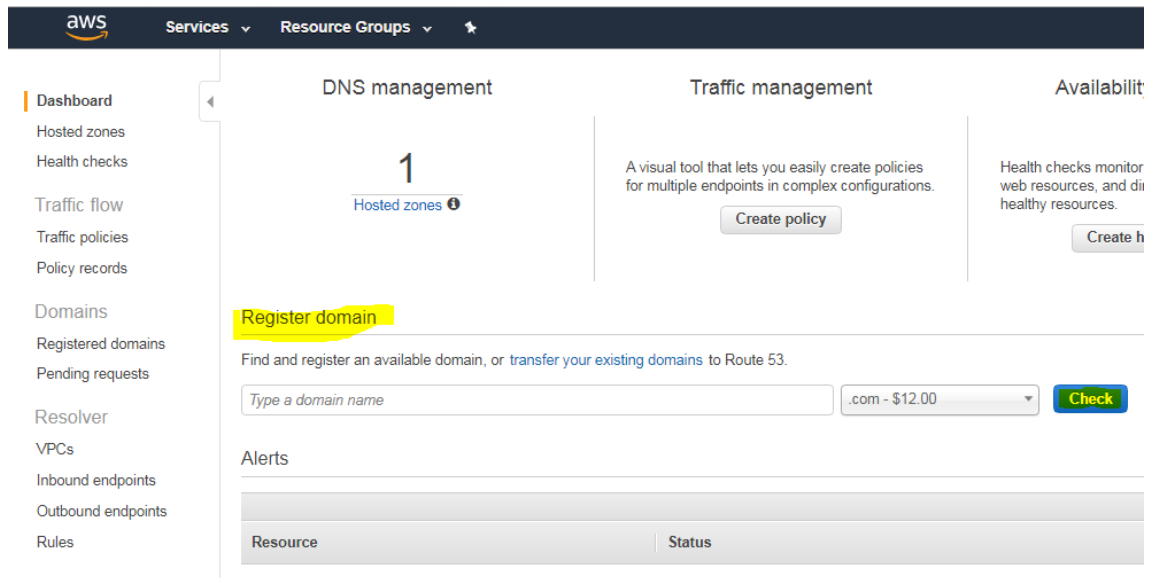


Figura 4.43. Configuración 2 de re-direccionamiento de la Web. Fuente: Pantallazo PC

En la siguiente imagen (Figura 4.44), vemos que se nos da información sobre si está o no disponible el dominio, se nos sugieren otros dominios y se nos da la opción de comprar el dominio. Una vez tendremos el dominio comprado siguiendo las instrucciones típicas de cualquier comprar por internet, tendremos que esperar si no recuerdo mal un máximo de 24h para poder utilizar el dominio.

Una vez el dominio esté listo para su utilización, ya podemos proceder a los pasos necesarios para vincularlo con los "Buckets" creados en AWS S3.

1: Domain Search

2: Contact Details

3: Verify & Purchase

Choose a domain name

richardtfg .com - \$12.00 [Check](#)

Availability for 'richardtfg.com'

Domain Name	Status	Price /1 Year	Action
richardtfg.com	✓ Available	\$12.00	Add to cart

Related domain suggestions

Domain Name	Status	Price /1 Year	Action
richardbmfg.com	✓ Available	\$12.00	Add to cart
richardsfg.com	✓ Available	\$12.00	Add to cart
richardsfg.net	✓ Available	\$11.00	Add to cart
richardtfg.info	✓ Available	\$12.00	Add to cart
richardtfg.me	✓ Available	\$17.00	Add to cart
richardtfg.net	✓ Available	\$11.00	Add to cart

Figura 4.44. Dominios disponibles y su coste. Fuente: Pantallazo PC

Con el dominio listo para su utilización en el mismo servicio de Route53 vamos a Hosted zones y ahí nos encontraremos con el dominio que hemos comprado, Figura 4.44.

Dashboard

Hosted zones

Health checks

Traffic flow

Traffic policies

Policy records

Domains

Registered domains

[Create Hosted Zone](#) [Go to Record Sets](#) [Delete Hosted Zone](#)

Search all fields X All Types

Domain Name	Type	Record Set Count	Comment
iotcloudsolutions.com	Public	4	HostedZone created by Route53 Registrar

Figura 4.45. Dominio comprado. Fuente: Pantallazo PC

Hacemos clic en el nombre del dominio comprado y nos encontramos con una imagen parecida a la de la Figura 4.45. Debemos darle a la opción “Create Record Set”, que nos permitirá vincular el dominio con los buckets creados.

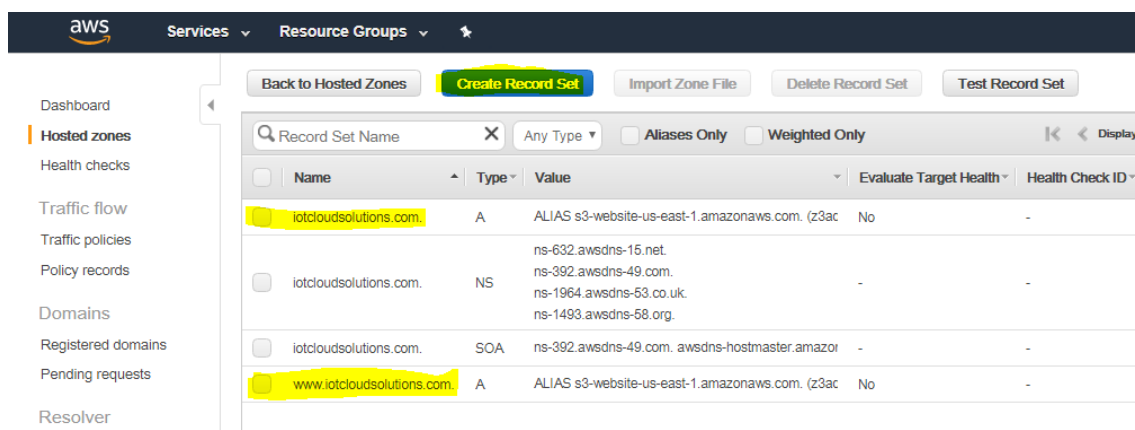


Figura 4.46. Asociar bucket creado con el Dominio comprado. Fuente: Pantallazo PC

Al hacer clic en “Create Record Set”, nos encontraremos con la imagen de la Figura 4.46, donde se nos pide un nombre del record set, como queremos que las personas al poner el nombre de dominio `iotcloudsolutions.com` entren a la web, no pondremos nada en ese espacio, es decir, lo dejaremos en blanco. En Alias, nos sale por defecto “NO”, podremos “YES” y eso nos permitirá ir a buscar el bucket en AWS S3 con el nombre `iotcloudsolutions`, en el caso de que hayamos dado el mismo nombre al dominio que a nuestra “Bucket” en AWS S3. Si observamos la imagen lo veremos más claro.

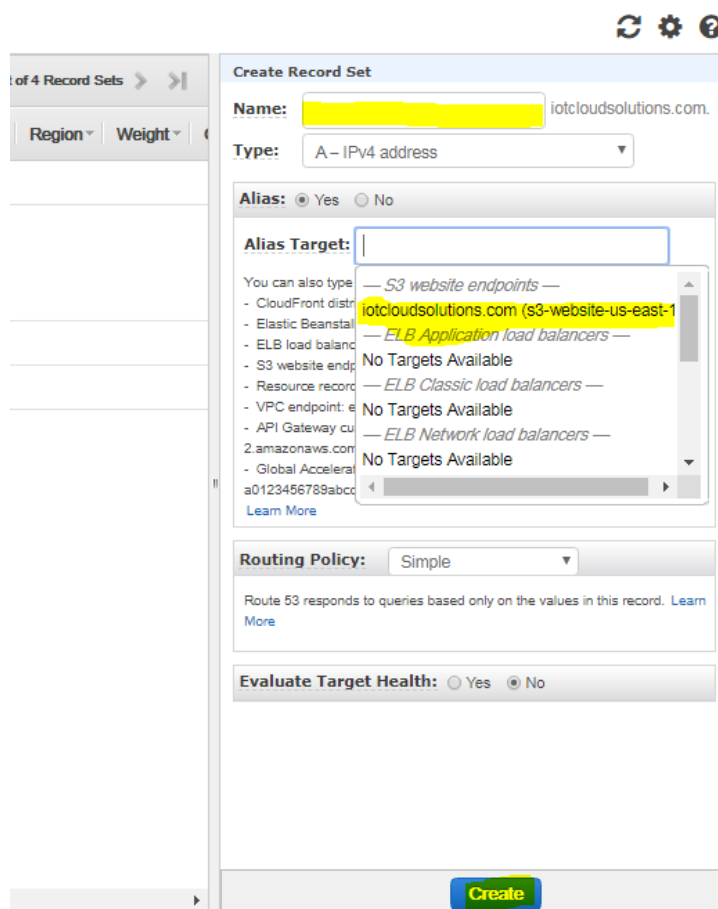


Figura 4.47. Asociación del bucket `iotcloudsolutions.com`. Fuente: Pantallazo PC

Una vez hayamos seguidos los pasos anteriores le damos a “Create” y ya lo tendremos hecho.

Ahora volveremos a crear otro “Record Set”, pero esta vez pondremos WWW donde pone Name, y lo asociaremos al bucket `www.iotcloudsolutions.com`. tal y como hicimos anteriormente.

The screenshot shows the AWS Route 53 console's 'Create Record Set' form. On the left, a table displays existing record sets with columns for Name, ID, TTL, Region, and Weight. The main form on the right is titled 'Create Record Set'. The 'Name' field contains 'www' and the 'Type' is set to 'A - IPv4 address'. The 'Alias' option is selected as 'Yes'. Under 'Alias Target', a dropdown menu is open, showing various target types. The target 'www.iotcloudsolutions.com (s3-website-us-east-2.amazonaws.com)' is highlighted. Other options include 'S3 website endpoints', 'ELB Application load balancers', 'ELB Classic load balancers', and 'ELB Network load balancers', each with a 'No Targets Available' message. The 'Routing Policy' is set to 'Simple', and 'Evaluate Target Health' is set to 'No'. A 'Create' button is at the bottom right.

Figura 4.48. Asociación del bucket `www.iotcloudsolutions.com`. Fuente: Pantallazo PC

Una vez hayamos hecho estos dos sencillos pasos, normalmente ocurre inmediatamente, pero a veces hay que esperar alrededor de una hora para que podamos acceder a nuestra página web poniendo el dominio que hemos comprado.

Con esto ya tendríamos la página web con el Dominio que hemos comprado.

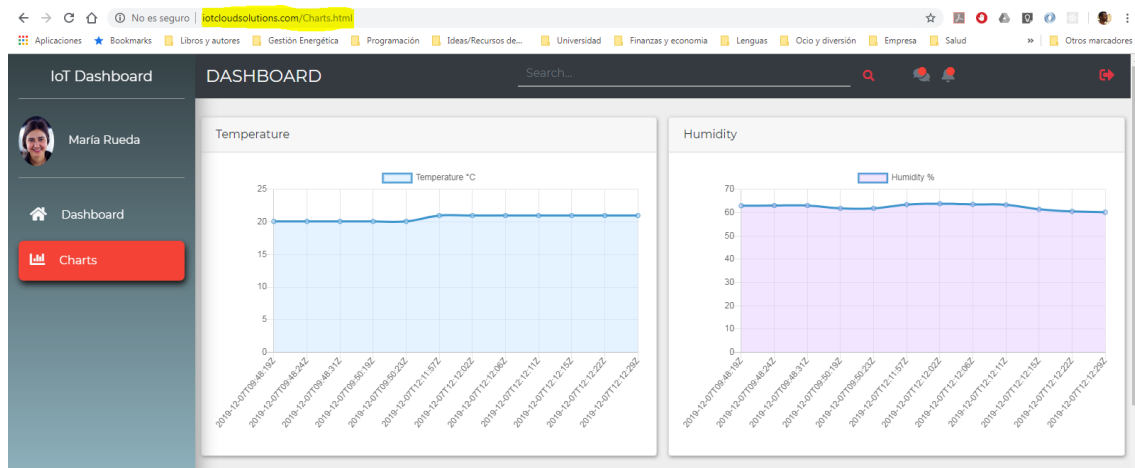


Figura 4.49. Resultado final arquitectura básica. Fuente: Pantallazo PC

Como podéis observar el documento se hace largo, para una arquitectura tan simple, una vez se sabe cómo crear esta parte simplificada, a continuación, se procede a escalar la plataforma y aumentar el número de sensores.

5. Coste de creación de la arquitectura básica.

En este apartado vamos a desglosar los costes y explicar cuanto nos ha costado realmente construir esta arquitectura básica.

Tabla 1. Coste arquitectura básica. Elaboración propia

COSTES ARQUITECTURA BÁSICA	
COSTE DE MATERIAL (HARDWARE)	
Nombre material	Precio
ABOX Raspberry Pi 3 B+ Starter Kit con Micro SD de 32GB Clase 10, 5V 3A Adaptador de Corriente con Interruptor, 2 Radiadores, Cable HDMI, Caja de Cali. [Comprado en Amazon sin costes de envío]	79,99 €
FULARR® 120Pcs Premium 20cm Dupont Cable, Breadboard Jumper Wire Kit, 40 Pines Macho-Macho, 40 Pines Hembra-Hembra, 40 Pines Macho-Hembra Dupont Puen. [Comprado en Amazon]	7,95 €
2 de Modulo DHT-22 Sensor Digital de Temperatura y Humedad AM2302 para Arduino. [Comprado en Amazon sin costes de envío]	6,24 €
Cable de Red Ethernet. CAT5e Networking cable 1 Gbps 2 m. [Comprado en FNAC]	9,99 €
Placa Prototipo MB-102 MB102 PROTOBOARD 830 Puntos. Ya disponible en dependencias [Precio Amazon]	3,10 €
TOTAL COSTES MATERIAL	107,27 €
COSTE SOFTWARE (LA NUBE, DOMINIO,...)	
Dominio en AWS Route53	12,00 €
Factura AWS mes de Octubre [Errores de novato]	6,60 €
Factura AWS mes de Noviembre [Errores de novato]	13,38 €
Factura AWS mes de Diciembre [Errores de novato]	6,21 €
TOTAL COSTES SOFTWARE	38,19 €

COSTES TOTALES ARQUITECTURA BÁSICA	145,46 €
---	-----------------

Como se puede ver en la tabla, el Software es lo más barato de construir toda la prueba de concepto, cosa que es normal.

Los gastos que tenemos por software deberían haber sido mucho menores, pero se han cometido errores utilizando el servicio de AWS, concretamente el servicio que nos ha costado más caro es DynamoDB, no porque no sea gratis o no entre en la capa gratuita sino porque la capa gratuita tiene unas limitaciones que nosotros hemos superado.

Lo más difícil de comprender en el servicio de AWS es el coste y saber predecir el coste de cada servicio. Por lo que hay que tomarse el tiempo para entender correctamente el coste de utilización de cada servicio.

En este caso con DynamoDB, hemos pagado por superar la capacidad mensual lectura y escritura. Al experimentar con este servicio de DynamoDB había creado muchas bases de datos que no estaba utilizando realmente, pero cada base de datos crear por defecto una capacidad de lectura y escritura de 5 diaria. Si multiplicas los 30 días del mes, por las 5 o 6 bases de datos que había creado y no utilizaba, se superó los límites de lectura y escritura de la capa gratuita.

Aun así, el riesgo estaba bastante controlado y no hemos llegado a pagar mucho más.

Para no incurrir en este tipo de imprevistos se recomienda, eliminar las bases de datos que no utilicemos, y configuremos la capacidad de lectura y escritura para que estén al mínimo si es posible, es decir a 5 o a 1. Y tener en cuenta que una vez superado las capacidades de lectura y escritura de la capa gratuita tendremos que pagar por este servicio. Con una base de datos o dos raras veces se superan los límites.

6. Arquitectura definitiva con más sensores.

Ahora que ya sabemos cómo construir una arquitectura básica de principio a fin vamos a escalar esa misma arquitectura poniendo más sensores.

En este apartado ya no se muestra un paso a paso, simplemente se muestra lo siguiente:

- Diagrama del circuito eléctrico.
- Diagrama de la arquitectura
- Imagen del resultado final / Link al resultado final
- Presupuesto de la arquitectura definitiva
- Enlace al código completo a través de Github

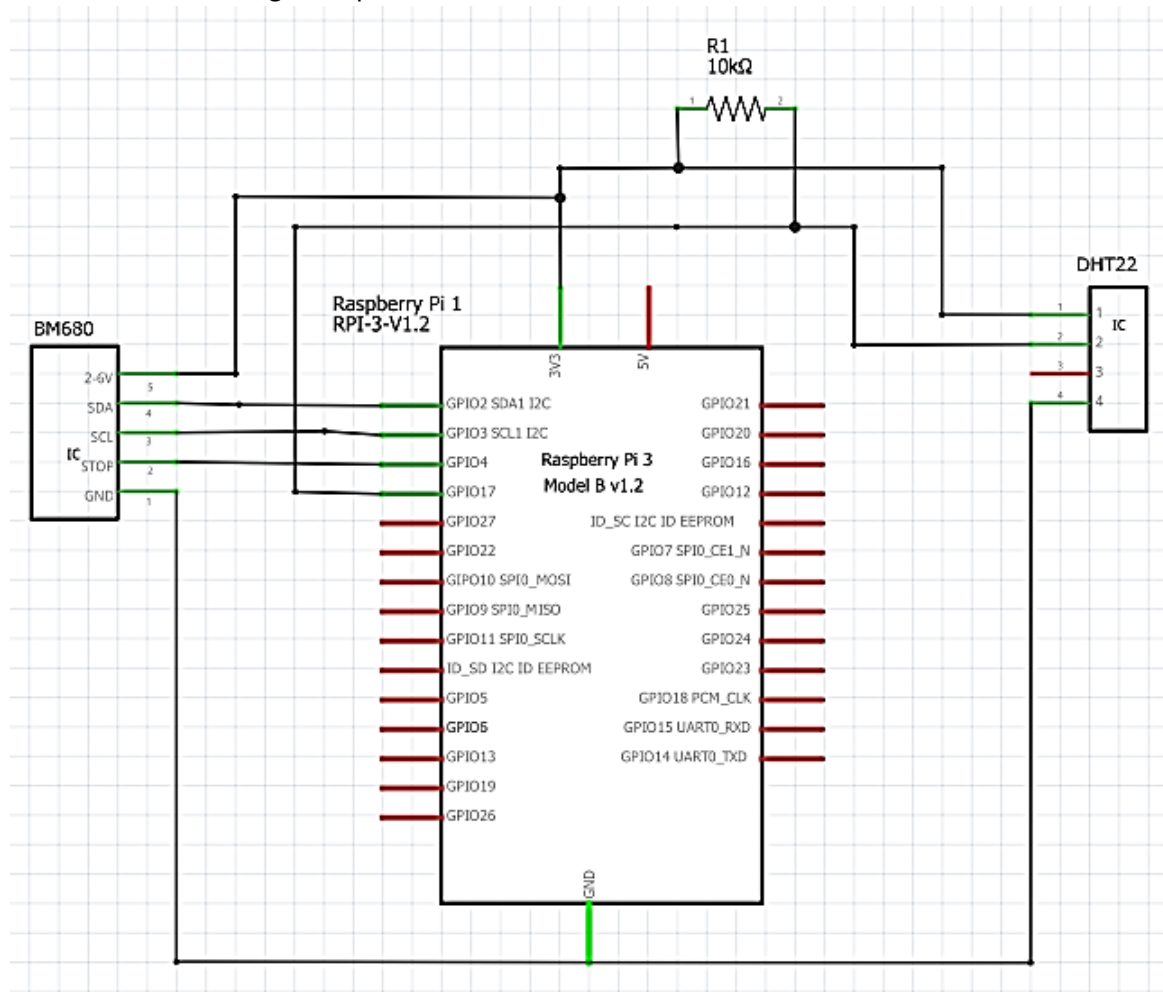


Figura 6.1. Circuito definitivo. Fuente: Elaboración propia

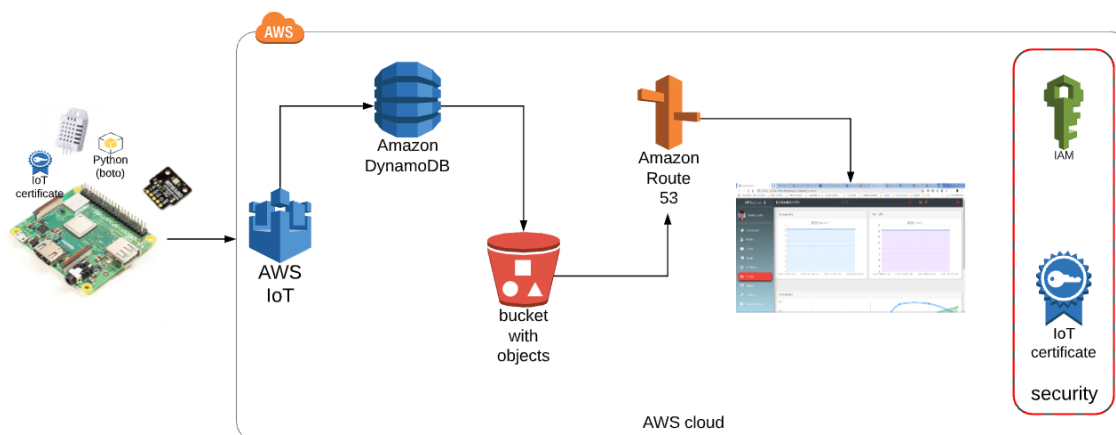


Figura 6.2. Arquitectura definitiva. Fuente: elaboración propia

Como se puede ver en la Figura 6.2, solo se ha incluido únicamente el sensor BME680, debido a los contratiempos con los que nos hemos encontrado.

En la Tabla 2, vemos que el coste total ha aumentado ya que ha tenido que comprar cables, enchufes y demás dispositivos Hardware.

También en la misma Tabla se puede ver que los costes de la nube, se mantienen igual. Con la nube de AWS, solo pagas por lo que utilizas y en relación a las experiencias pasadas se han utilizado los servicios adecuadamente para no incurrir en gastos extras o no esperados.

Tabla 2. Coste arquitectura definitiva. Elaboración propia

COSTES ARQUITECTURA BÁSICA	
COSTE DE MATERIAL (HARDWARE)	
Nombre material	Precio
ABOX Raspberry Pi 3 B+ Starter Kit con Micro SD de 32GB Clase 10, 5V 3A Adaptador de Corriente con Interruptor, 2 Radiadores, Cable HDMI, Caja de Cali. [Comprado en Amazon sin costes de envío]	79,99 €
FULARR® 120Pcs Premium 20cm Dupont Cable, Breadboard Jumper Wire Kit, 40 Pines Macho-Macho, 40 Pines Hembra-Hembra, 40 Pines Macho-Hembra Dupont Puen. [Comprado en Amazon]	7,95 €
2 de Modulo DHT-22 Sensor Digital de Temperatura y Humedad AM2302 para Arduino. [Comprado en Amazon sin costes de envío]	6,24 €
Adafruit TSL2561 High Dynamic Range Digital Light Sensor	13,49 €
SDM120CT-MV + ESCT-TU16 50A/0,1 V 1 de la Fase 2 de 230V RS485 Modbus y medida de kwh kvarh U-P Q PF Hz dmd medidor de energía de carril DIN. [Gastos de envío Fedex incluido]	26,34 € + 43,00 € (Envío)
Cable de Red Ethernet. CAT5e Networking cable 1 Gbps 2 m. [Comprado en FNAC]	9,99 €
Cable eléctrico de tres hilos (tierra, neutro y fase) de sección nominal de 1 mm².	11,15 €
Base De Enchufe Estanca Con Tapa Para Superficie, Gris de Legrand.	9,99 €
Cable de alimentación Cable de alimentación con interruptor enchufe tipo C a abierto Cable Extremos Longitud: 1,50 m.	6,50 €
ARCELI USB a 485 a 485 convertidor USB a RS485 USB a Puerto Serie 485 Compatible con win8 win7.	5,99 €
BME680 Breakout - Air Quality, Temperature, Pressure, Humidity Sensor	20,75 €
Placa Prototipo MB-102 MB102 PROTOBOARD 830 Puntos. Ya disponible en dependencias [Precio Amazon]	3,10 €
TOTAL COSTES MATERIAL	244,48 €
COSTE SOFTWARE (LA NUBE, DOMINIO,...)	
Dominio en AWS Route53	12,00 €
Factura AWS mes de Octubre [Errores de novato]	6,60 €
Factura AWS mes de Noviembre [Errores de novato]	13,38 €
Factura AWS mes de Diciembre [Errores de novato]	6,21 €
TOTAL COSTES MATERIAL	38,19 €
COSTES TOTALES ARQUITECTURA BÁSICA	282,67€

7. 5. Conclusiones de la prueba de concepto (PdC)

Como en todo proyecto experimental, una persona, por mucho que defina el resultado deseado, resulta difícil dar con él si es un iniciado en las áreas de conocimiento que involucran el proyecto. Crear una plataforma de transmisión de datos de sensores en tiempo real y con unas características de presentación según tus propios deseos con unas tecnologías y lenguajes de programación diversos ha sido todo un reto. A continuación, explico algunos de los aprendizajes que me he llevado realizando esta fase experimental:

- Utilizar la Raspberry Pi en modo headless, solo utilizando la línea de comandos puede parecer difícil e intimidante... Pero con el tiempo resulta fácil, ya que solo se utilizan un par de comandos para moverse entre carpetas e instalar paquetes.
- Existen muchas tecnologías para realizar una prueba de concepto parecida a la mostrada en este documento. Se puede optar por utilizar otros lenguajes de programación tanto para la parte de presentación como para el Back-en, lo mejor es enfrentarte a un lenguaje que te gustaría aprender o con el que te sientes cómodo utilizando.
- La nube de AWS es relativamente fácil de utilizar. A base de clics, se pueden hacer muchas cosas complejas. Mejor dicho, los servicios que se han utilizado son fáciles, lo más complicado es entender los precios de los diferentes servicios y cómo utilizarlo para pagar lo justo y necesario.
- No es necesario tener un conocimiento profundo de ninguna de las capas que componen el concepto de IoT, para realizar cualquier proyecto.
- Hablando de costes, lo más caro es el material inicial, en los presupuestos se puede ver que lo más caro ha sido el kit con la Raspberry Pi, la gran mayoría de los componentes han costado alrededor de 10 €.
- El tiempo de ejecución de las pruebas de concepto podría reducirse a 3 días o como mucho 1 semana si se trabaja bastante el diseño y la presentación. Si se parte de unos conocimientos básicos, puede llevar meses de aprendizaje.
- Es conveniente utilizar aparatos de protección frente a cortocircuitos y/o contactos directos e indirectos (fusibles, magnetos, ViGi, etc) si se desea proteger los aparatos de mayor coste.

8. Recursos utilizados para crear la prueba de concepto.

Para resolver gran parte de los retos con los que me he encontrado para realizar esta simple prueba de concepto he utilizado blogs, tutoriales de Youtube e incluso cursos online. A continuación, paso a poner algunos de los enlaces que me han ayudado a encajar los puzles (la nube, programación, raspberry pi, etc):

Raspberry Pi:

Curso Udemy: Tech Explorations™ Raspberry Pi Full Stack Raspbian

<https://www.raspberrypi.org/downloads/raspbian/>

<http://frederickvandenbosch.be/?p=2385>

<https://raspberrypi.hq.com/how-to-connect-your-raspberry-pi-to-wifi/>

<https://pinout.xyz/>

<https://www.raspberrypi.org/documentation/usage/gpio/>

<https://www.youtube.com/watch?v=eV3YHiEV5eQ>

<https://www.youtube.com/watch?v=wyKxOC2YOrU>

<https://www.youtube.com/watch?v=BxqnhEqPyGg>

La nube de AWS

Curso Udemy: AWS Certified Solutions Architect - Associate 2020

https://aws.amazon.com/es/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&all-free-tier.q=cloud9&all-free-tier.q_operator=AND

<https://www.youtube.com/watch?v=clvMR4kzcxc>

<https://medium.com/coinmonks/architecting-your-iot-app-using-raspberry-pi-and-aws-b8b89e3ac39a>

<https://hackernoon.com/building-an-iot-dashboard-using-the-onion-omega-and-amazon-aws-a3520f850c9>

<https://aws.amazon.com/es/getting-started/projects/build-serverless-web-app-lambda-api-gateway-s3-dynamodb-cognito/>

https://docs.aws.amazon.com/es_es/iot/latest/developerguide/iot-ddb-rule.html

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/dynamodb.html>

<https://stackoverflow.com/questions/41145694/update-chartjs-line-chart-from-dynamodb-query>

https://docs.aws.amazon.com/es_es/amazondynamodb/latest/developerguide/GettingStarted.NodeJs.04.html

Anexos

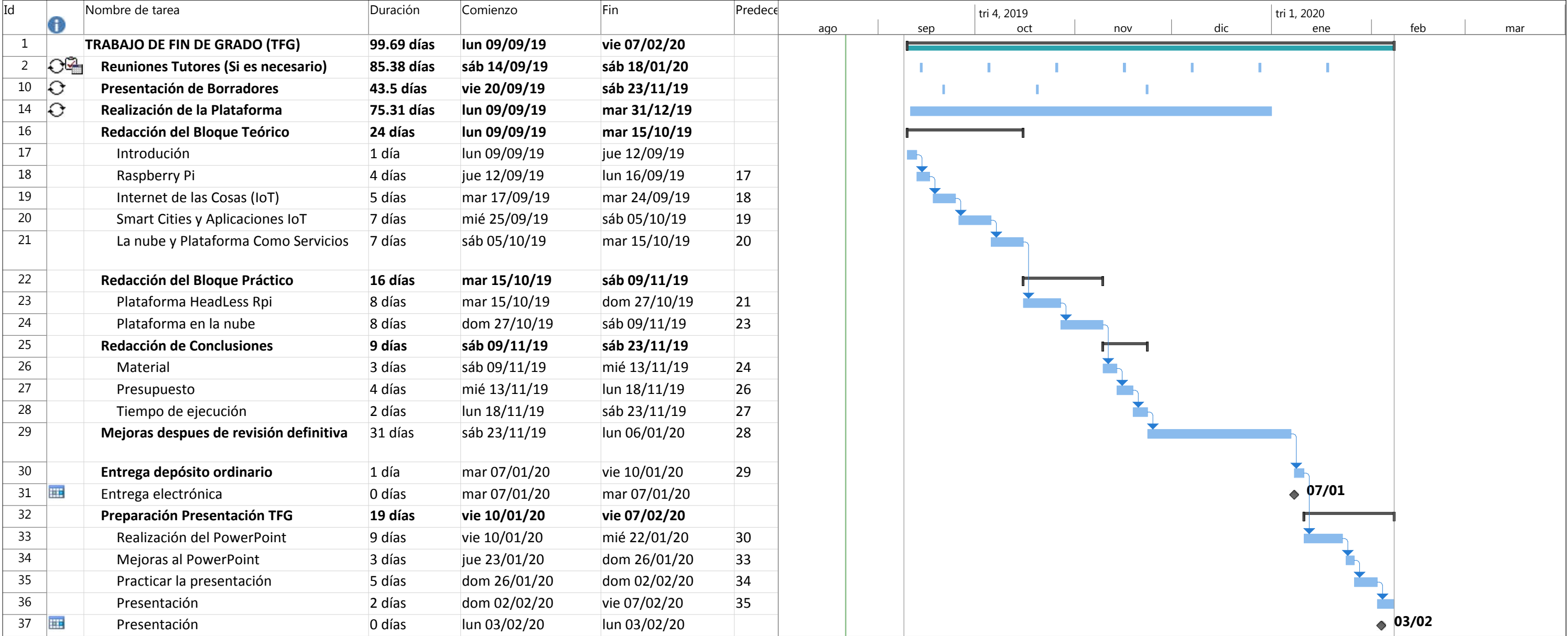
I. Planificación de TFG (Diagrama Gant)

II. Desglose de costes AWS

III. Reportaje fotográfico

I. Planificación de TFG (Diagrama Gant)

El diagrama de Gant que se muestra a continuación, es un diagrama que se hizo al inicio de comenzar el proyecto y por muy sofisticado que pueda parecer, no tiene nada que ver con la realidad. Debido a los contratiempos que me he encontrado tanto personales como los relativos al proyecto en sí. Ambos difíciles de predecir han hecho que sea imposible seguir con la planificación que se muestra en el diagrama.



Proyecto: Proyecto1.mpp
Fecha: jue 22/08/19

Tarea		Resumen del proyecto		Tarea manual		solo el comienzo		Fecha límite	
División		Tarea inactiva		solo duración		solo fin		Progreso	
Hito		Hito inactivo		Informe de resumen manual		Tareas externas		Progreso manual	
Resumen		Resumen inactivo		Resumen manual		Hito externo			

II. Desglose de costes AWS



Account number:

Bill to Address:
ATTN:

Request For Payment

Email or talk to us about your AWS account or bill, visit console.aws.amazon.com/support/

Bill Summary

Bill Number:

Bill Date: October 3 , 2019

TOTAL AMOUNT DUE ON October 3 , 2019

\$6.60

This bill is for the billing period September 1 - September 30 , 2019

Greetings from Amazon Web Services, we're writing to provide you with a bill for your use of AWS services. Additional information about your bill, individual service charge details, and your account history are available on the Account Activity Page.

Summary	
AWS Service Charges	\$6.60
Charges	\$5.45
Credits	\$0.00
Tax *	\$1.15
Total for this bill	\$6.60

Detail	
AWS IoT Analytics	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon Simple Storage Service	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS Data Transfer	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon S3 Glacier Deep Archive	\$0.00
Charges	\$0.00
VAT **	\$0.00
AmazonCloudWatch	\$0.12

** Related VAT invoices can be accessed by going to the Bills page on your Billing Management Console

Service Provider:
(Not to be used for payment remittance)
Amazon Web Services EMEA SARL
38 avenue John F. Kennedy,
L-1855 Luxembourg

Charges	\$0.10
VAT **	\$0.02
Amazon QuickSight	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS IoT	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon DynamoDB	\$5.87
Charges	\$4.85
VAT **	\$1.02
Amazon Elastic Compute Cloud	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS Key Management Service	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS Glue	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon Route 53	\$0.61
Charges	\$0.50
VAT **	\$0.11

** Related VAT invoices can be accessed by going to the Bills page on your Billing Management Console

Service Provider:
(Not to be used for payment remittance)
Amazon Web Services EMEA SARL
38 avenue John F. Kennedy,
L-1855 Luxembourg



Account number:

Bill to Address:
ATTN:

Request For Payment

Email or talk to us about your AWS account or bill, visit console.aws.amazon.com/support/

Bill Summary

Bill Number:

Bill Date: November 3 , 2019

TOTAL AMOUNT DUE ON November 3 , 2019	\$13.38
--	----------------

This bill is for the billing period October 1 - October 31 , 2019

Greetings from Amazon Web Services, we're writing to provide you with a bill for your use of AWS services. Additional information about your bill, individual service charge details, and your account history are available on the Account Activity Page.

Summary	
AWS Service Charges	\$13.38
Charges	\$11.06
Credits	\$0.00
Tax *	\$2.32
Total for this bill	\$13.38

Detail	
Amazon Simple Storage Service	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS Lambda	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS Data Transfer	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon QuickSight	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS IoT	\$0.00

** Related VAT invoices can be accessed by going to the Bills page on your Billing Management Console

Service Provider:
(Not to be used for payment remittance)
Amazon Web Services EMEA SARL
38 avenue John F. Kennedy,
L-1855 Luxembourg

Charges	\$0.00
VAT **	\$0.00
Amazon Cognito	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon DynamoDB	\$12.29
Charges	\$10.16
VAT **	\$2.13
Amazon Elastic Compute Cloud	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS Glue	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon Route 53	\$0.61
Charges	\$0.50
VAT **	\$0.11
Amazon Simple Notification Service	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS IoT Analytics	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon S3 Glacier Deep Archive	\$0.00
Charges	\$0.00
VAT **	\$0.00
AmazonCloudWatch	\$0.48
Charges	\$0.40
VAT **	\$0.08
AWS Key Management Service	\$0.00
Charges	\$0.00
VAT **	\$0.00

** Related VAT invoices can be accessed by going to the Bills page on your Billing Management Console

Service Provider:
(Not to be used for payment remittance)
Amazon Web Services EMEA SARL
38 avenue John F. Kennedy,
L-1855 Luxembourg



Account number:

Bill to Address:
ATTN:

Request For Payment

Email or talk to us about your AWS account or bill, visit console.aws.amazon.com/support/

Bill Summary

Bill Number:

Bill Date: December 3 , 2019

TOTAL AMOUNT DUE ON December 3 , 2019	\$6.21
--	---------------

This bill is for the billing period November 1 - November 30 , 2019

Greetings from Amazon Web Services, we're writing to provide you with a bill for your use of AWS services. Additional information about your bill, individual service charge details, and your account history are available on the Account Activity Page.

Summary	
AWS Service Charges	\$6.21
Charges	\$5.13
Credits	\$0.00
Tax *	\$1.08
Total for this bill	\$6.21

Detail	
Amazon Simple Storage Service	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS IoT Analytics	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS Data Transfer	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon S3 Glacier Deep Archive	\$0.00
Charges	\$0.00
VAT **	\$0.00
AmazonCloudWatch	\$0.18

** Related VAT invoices can be accessed by going to the Bills page on your Billing Management Console

Service Provider:
(Not to be used for payment remittance)
Amazon Web Services EMEA SARL
38 avenue John F. Kennedy,
L-1855 Luxembourg

Charges	\$0.15
VAT **	\$0.03
Amazon QuickSight	\$0.00
Charges	\$0.00
VAT **	\$0.00
AWS IoT	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon Elastic Compute Cloud	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon DynamoDB	\$5.42
Charges	\$4.48
VAT **	\$0.94
AWS Glue	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon Simple Notification Service	\$0.00
Charges	\$0.00
VAT **	\$0.00
Amazon Route 53	\$0.61
Charges	\$0.50
VAT **	\$0.11

** Related VAT invoices can be accessed by going to the Bills page on your Billing Management Console

Service Provider:
(Not to be used for payment remittance)
Amazon Web Services EMEA SARL
38 avenue John F. Kennedy,
L-1855 Luxembourg

III. Reportaje fotográfico

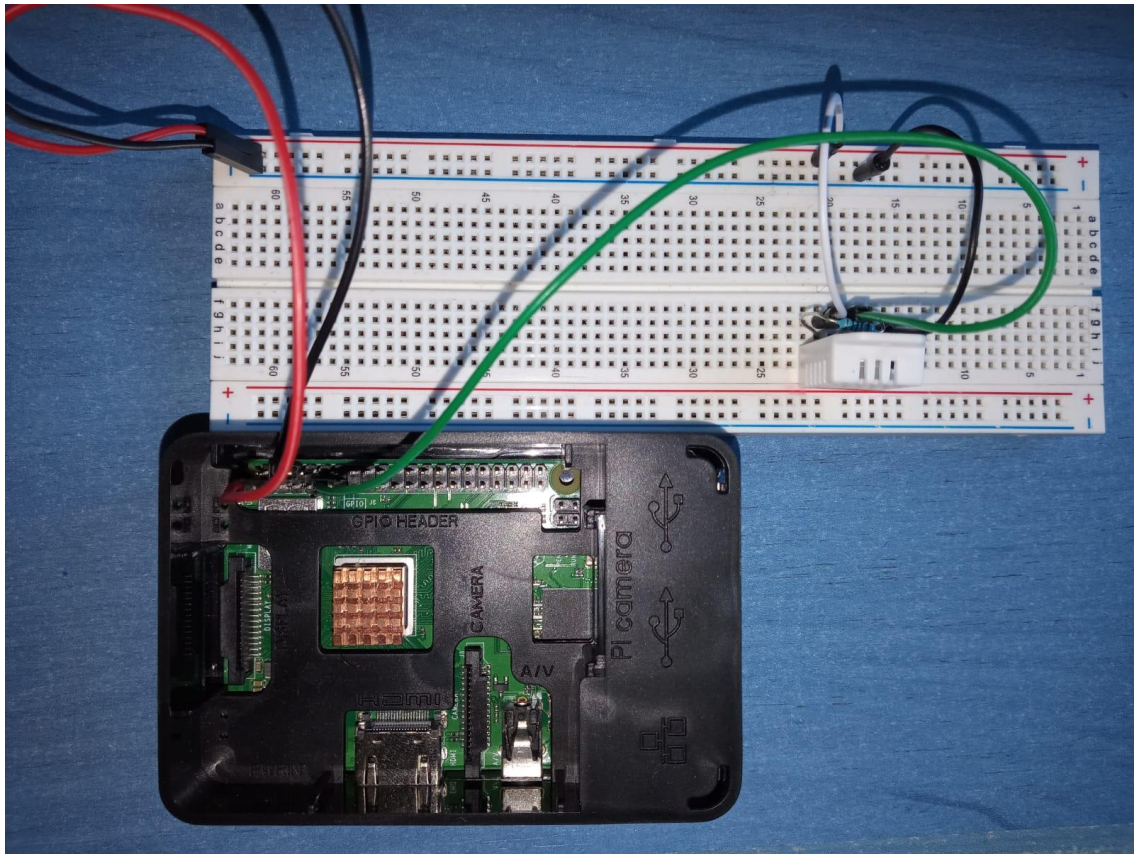


Figura 1. Raspberry Pi conectado al sensor DHT22



Figura 1. Sensor de TSL2561 (Luxómetro)

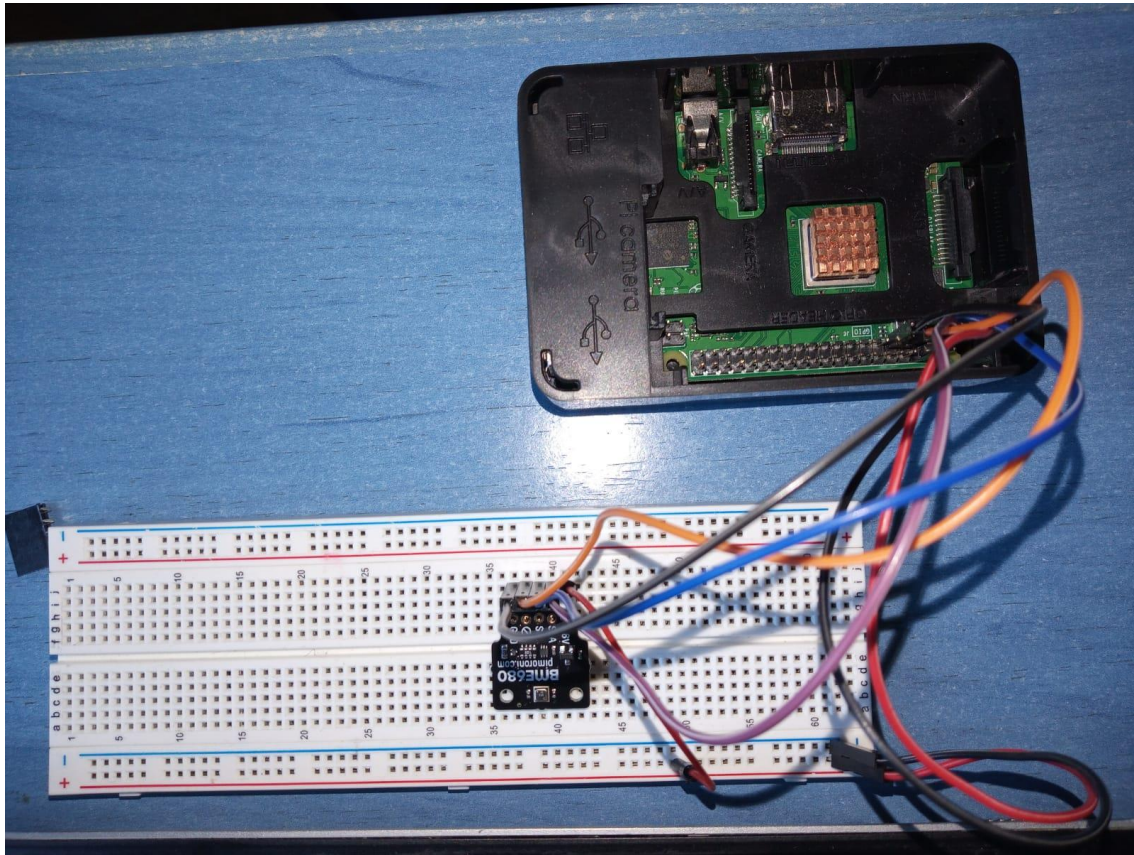


Figura 2. Raspberry Pi conectado al sensor BME680

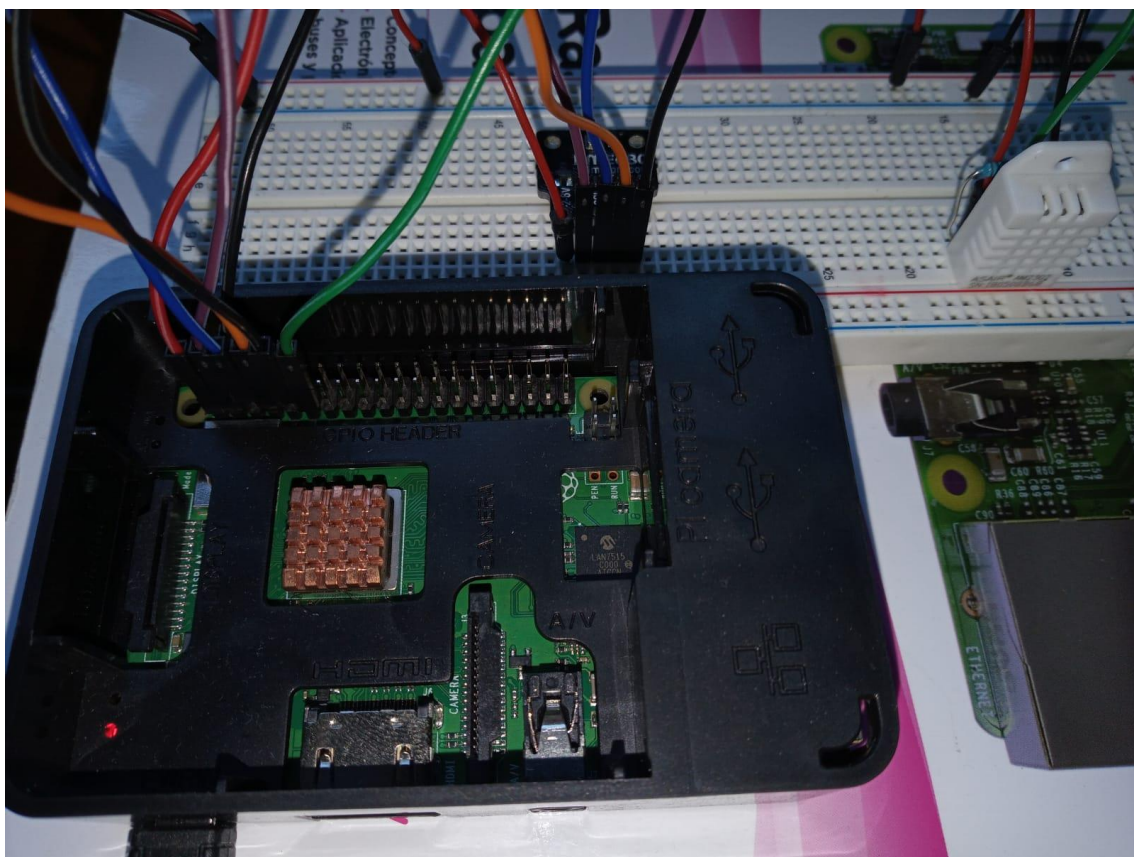


Figura 3. Raspberry Pi conectado a los dos sensores (DHT22 y BME680)

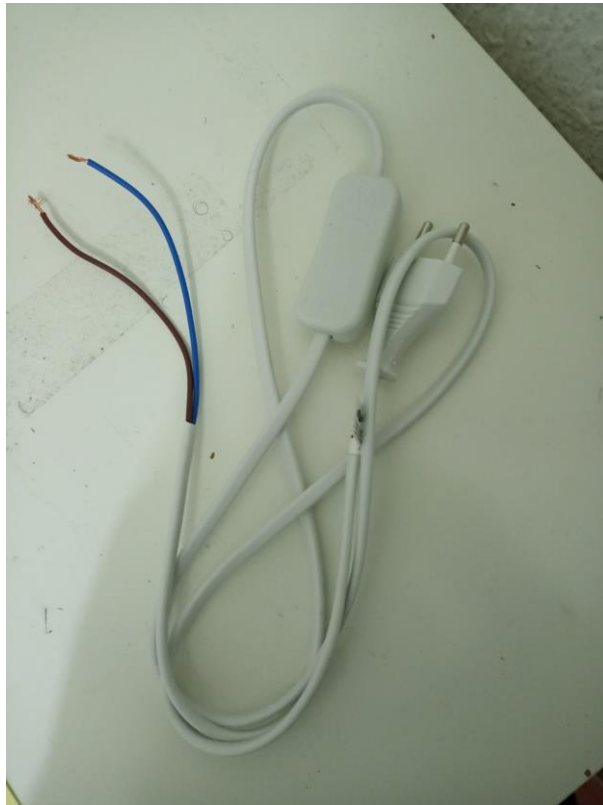


Figura 4. Cable alimentación con enchufe y extremo abierto



Figura 5. Accesorio convertidor USB a RJ485 (Comunicación Modbus con RPi)



Figura 6. Pinza amperimétrica y Medidor de energía SDM120CT estropeado

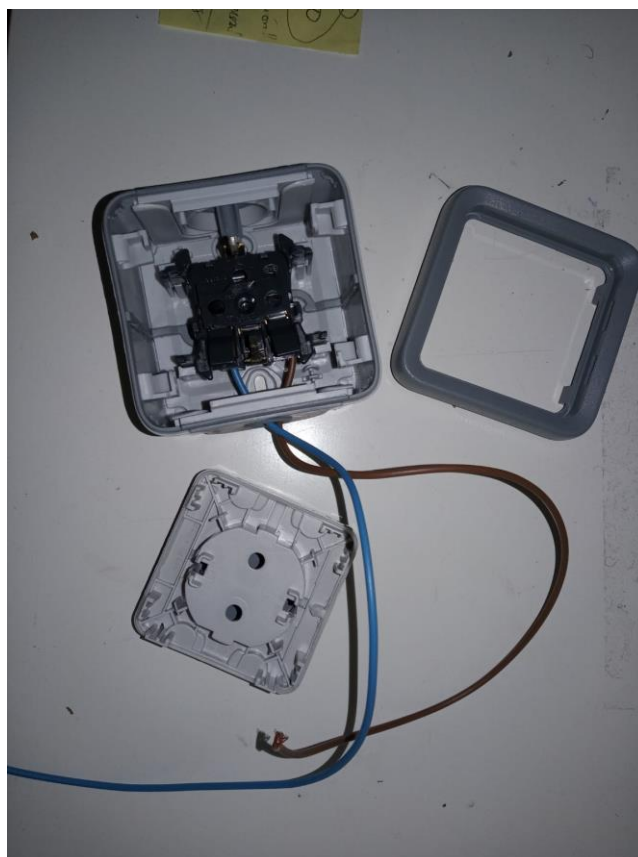


Figura 6. Base de enchufe desmontable